Case Study of a Low Power MTCMOS based ARM926 SoC : Design, Analysis and Test Challenges

Sachin Idgunji
ARM Inc. Sunnyvale, USA

Abstract

Low Power techniques such as multi-voltage islands, voltage scaling and power gating are gaining ground to address the need for managing energy and power during run time as well as during standby modes. These design techniques increase the complexity of implementing and analyzing the power network to meet the required average IR drop requirements , transient turn on in-rush (out-rush) currents as well as dynamic voltage drops during runtime.

We designed an ARM926 based SoC that implements the above techniques including a multi-threshold CMOS (MTCMOS) based processor core. To analyze the effects of effects associated with power gating we have built in, as part of the logic design , a system that uses the scan chain based network to observe the effects of transient currents on the virtual power network on state retention . The system can also be used to check the integrity of the retention latches when the design has been powered off using the MTCMOS network.

1. Introduction

Energy management and power dissipation of complex SoCs is increasingly becoming a key part of the design specifications. It is well understood that managing both the peak as well as average power dissipation will reduce the manufacturing and packaging costs as well as improve the reliability.

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

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

1.1 Dynamic Power Management

To minimize the dynamic power dissipation term the clock frequency (f_{clk}) needs to be lowered as well as the switching activity (α) and the supply voltage (V_{DD}) should be reduced.

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. Techniques such as Clock Gating (CG) can yield a significant saving in both power dissipation and energy consumption and are now common in design flows.

Dynamic Frequency Scaling (DFS), wherein the frequency of the system is scaled below maximum desired performance leads to a linear reduction in average power dissipation but does not reduce the energy consumption for a given task [2]. However an accompanying reduction in the supply voltage to a level that is just high enough to support this lowered clock frequency results in lesser energy consumption when charging the load capacitance. This technique, known as Dynamic Voltage and Frequency Scaling (DVFS), leads to a quadratic reduction in energy consumption [6].

1.2 Leakage Power Trends

Leakage power, a property that transistors are not perfect switches, has become significant with technology scaling because of lower threshold voltages, thinner oxides as well as high electric fields. Figure 1 shows the savings in dynamic energy, compared to 180nm, for different libraries at advanced feature nodes. These savings have been offset by accompanied increase in standby power due to leakage currents through the devices.

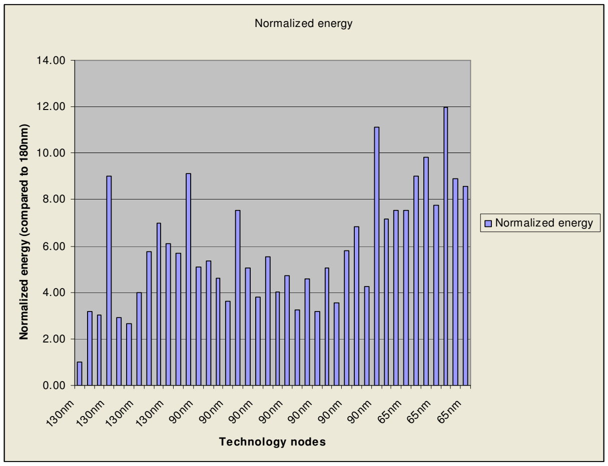
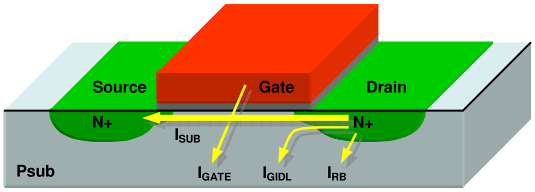


Figure 1 - Trends in Dynamic Energy savings in standard cell libraries (across foundries)

1-4244-1128-9/07/$25.00 © 2007 IEEE

Leakage power is dissipated in both active mode and standby mode and the currents which contribute to leakage are rising with each technology node. Even with low leakage processes, the increased power density (to meet performance needs) due to increased overdrive offsets the gains in lower leakage. Using the standard performance processes, in some applications it may be more energy efficient to run fast and stop/shutoff blocks of logic rather than to lower the voltage and frequency due to the high active leakage currents or use lower leakage processes with overdrive supply voltages.

The four main components of leakage current , sub-threshold Leakage (I_{SUB}), Gate Leakage (I_{GATE}), Gate Induce Drain Leakage (I_{GIDL}) and Reverse Bias Junction Leakage (I_{RB}) are increasing with each technology node [3] [5].



$$I_{LEAK} = I_{SUB} + I_{GATE} + I_{GIDL} + I_{RB} \qquad (2)$$

Figure 2 - Components of leakage current in an NMOS transistor

The above components of leakage current are also proportional to the supply voltage at which the system operates.

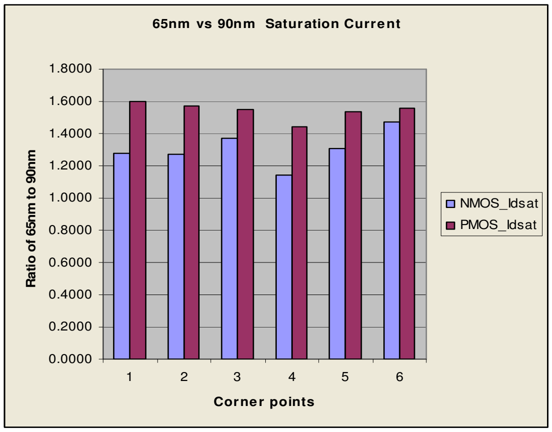


Figure 3 – Saturation current ratios at different voltage/temperature points

Figure 3 shows the increase in saturation as measured in silicon at different “voltage/temperature” points. The average increase of 30% in saturation current is offset by the increase in 500% in leakage current (all components) measured at the same corner points, as shown in Figure 4.

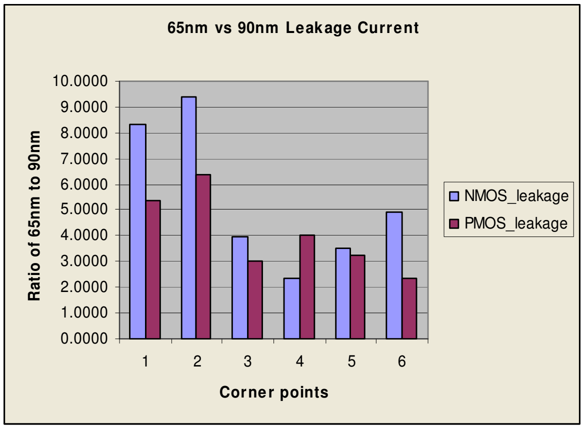


Figure 4 – Leakage current ratio at the same voltage/temperature points used to measure saturation current

For the same corner points, we measured the gate leakage for an array of 0.4um width NMOS transistors. This showed an average increase of 21% in gate leakage across the corner points (same as in Figure 3 and 4). This component of leakage is sensitive to the supply voltage at which the circuit operates

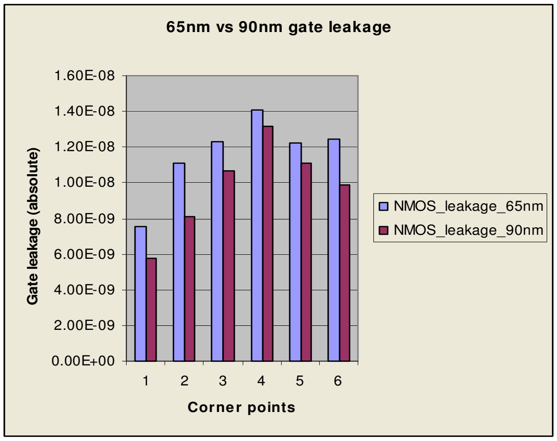


Figure 5 – Average gate leakage in a 0.4u NMOS

To manage both leakage as well as dynamic power, an ARM926 based system was developed, that included dynamic energy management techniques such as scaling frequency (DFS) as well as limited voltage scaling (DVFS), and also focused on managing leakage using a variety of techniques.
These techniques were applied in system during runtime such as when decoding an MPEG stream. Some of the observations from the system are captured further along in the paper.

This system was targeted to a 90nm standard performance process.

2 ARM926 based SoC – SALT

The **S**ynopsys **A**RM **L**eakage **T**echnology demonstrator known as "SALT" was an R&D collaboration between ARM and Synopsys to explore the practical details of implementing some of the more aggressive leakage mitigation techniques such as MTCMOS power gating [1] [4], dual V_T optimization and VTCMOS as these techniques are effective at managing leakage power.

2.1 SALT Architecture and Design Characteristics

The design of the SALT technology demonstrator was to build on an established ARM926EJS based sub-system with the addition of an intelligent leakage controller for managing the different power management modes in the system. The overall SoC was designed to support 3 voltage domains – 2 within the ARM926EJS system and 1 for the logic in the SoC subsystem. An OTG USB core in the SoC subsystem, from Synopsys, was also implemented with MTCMOS headers. The ARM926EJS was partitioned in to two voltage domains- VDDCPU which supplied power to the CPU logic cells and VDDRAM which supplied power to the memories in the processor core, to allow the RAMs and any associated logic like the isolation cells 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 rail collapse on the power supply network during start up, when the sleep signal to the MTCMOS switches was de-asserted. At nominal speed of 300MHz, the CPU operated at 3x of the AHB subsystem frequency. However the subsystem frequency could be increased to 133 MHz. To support dynamic frequency scaling, the processor clock could be run at 4x, 2x or at the same frequency as the AHB clock. The dynamic clock generator logic could be programmed by the energy controller block.

Besides the DFS and DVFS modes which could be applied to manage dynamic power, the SALT system also included the following 5 modes to manage leakage power.

1. **Halt** – simple stopping of the clocks. This mode also represents the baseline against which the other modes are compared.
2. **Halt with VDD scaling -** Clock gating with clamping/isolation and VDD scaling to lower leakage.
3. **Light Sleep** – the CPU is power gated and the state retained in retention registers.
4. **Deep Sleep** – the CPU is switched off and the register states are retained in RAM
5. **Shutdown** – both CPU and RAM are switched off and logical state retained in SoC RAM or external Flash.

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 in the paper.

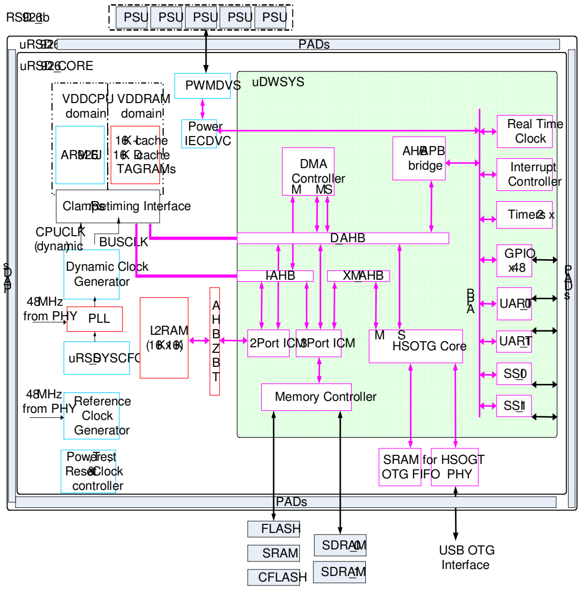


Figure 6 – SALT design and architecture

All the peripheral subsystem with the exception of the High Speed OTG core, which could be power gated, operated on an always-on 1.0V nominal supply. Although the design was not implemented with level shifters, the CPU and the RAM voltage domains could be scaled to achieve limited DVFS.

The design was targeted to an experimental "R&D" library based on ARM's SAGE-X standard cell library in a 90nm process. To support VTCMOS each cell was tap-less and special well tap cells were added that were connected to bias supplies that could modulate the well bias voltage during runtime.
A deep nwell was added to isolate "pwells" and back-bias the NMOS transistors. An extra 10th track supplying true V_{DD} was added 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

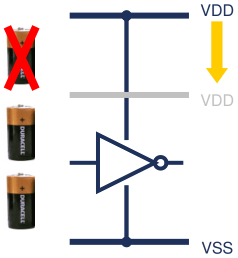
a power management kit consisting of the following cells was also created, drawn to the same standard cell rules:

- **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

3 Leakage mitigation techniques implemented in the SALT design

Of the several techniques available to reduce leakage, the following were implemented in the SALT design. Some of these such as Dual V_T and VTCMOS rely on additional support in the manufacturing process to lower the leakage and ones like the Dual Vt are static implemented during optimization phase of the design flow whilst others such as Power Gating and Stack Effect are stand alone circuit techniques.

3.1 Lowering Supply Voltage



Referring to equation (1) it can be seen that leakage power will reduce linearly with the lowering of the supply voltage (V_{DD}) however any reduction in V_{DD} also reduces the MOSFET gate drive (V_{GS}-V_T). It can be seen from equation 2 below, that a reduction in supply voltage (Vds) can reduce the sub-sub-threshold component of leakage current as well.

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

The figure below shows the impact of lowering VDD during the standby mode (mode #2) which shows a significant reduction in leakage current (compared to mode #1)

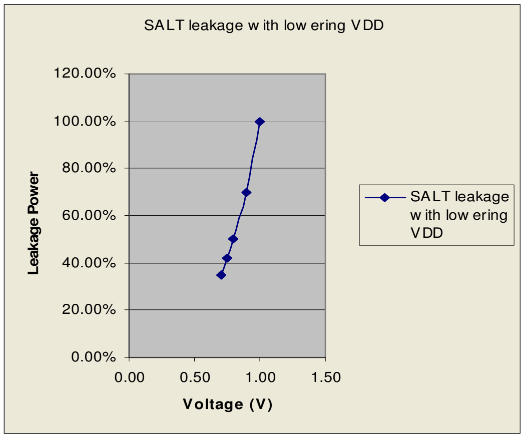
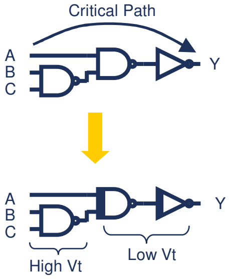


Figure 7- Leakage on SALT silicon with reduction in voltage

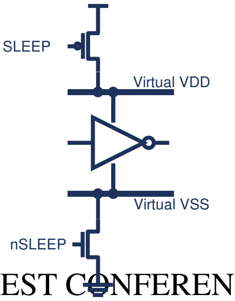
3.2 Multi (Dual) V_T optimization



The SALT implementation used a dual V_T library during synthesis (logical and physical) 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

Using multi-Vt synthesis reduces leakage at the expense of the overall slack in the design, the side effect of this is that as more number of timing paths get closer to the critical path, the statistical probability of paths failing to meet required performance target increases. If, however minimizing leakage has a higher cost priority then this process can be done aggressively by decreasing the design slack range, performing high leakage equivalents cells in speed critical paths.
In the SALT design, the total design slack was reduced by 40% of the original slack.

3.3 Power Gating



A far more aggressive and effective technique for leakage mitigation is to simply cut the power supply to any inactive transistor.
This is done by placing MOS switches in the power network,

the ground network or both. The exact placement and sizing of these switches must be done to avoid an adverse impact on performance.

The SALT design implemented header only switches to implement power gating. The motivation for using the header switch design was the active high nature of processor control signals and using an implementation that integrated seamlessly with the limited DVFS that we wanted to implement. For the header switch design, after much simulation it was decided that a switch transistor of width 0.55µm and length 0.13µm provided the best R_{ON} to I_{OFF} ratio. The switch cells were built out multiple transistor fingers of this size in parallel. Each cell comprised of 30 transistors.

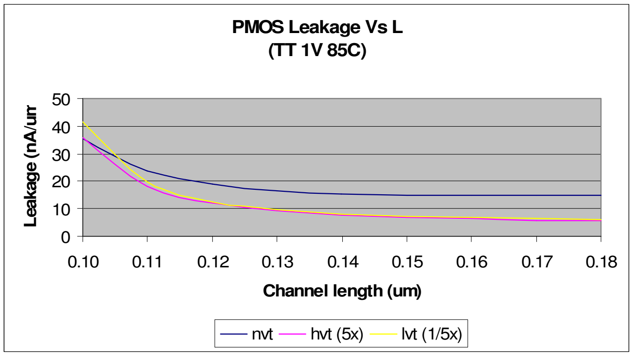


Figure 8 - Leakage in header cells with varying transistor length

SPICE simulations were run on a representative test circuit with varying number of headers and the load that they were supplying to study the effects on signal delay, IR drop and leakage. It was found that spacing the switches every 50µm resulted in less than a 5% IR drop from the power supply for a load that operated at 350MHz.

The power gates(header switches) were laid out as double height cells and stacked in columns with all the pin connectivity done by abutment in the placement area where the synthesized CPU logic was placed (VCPU).

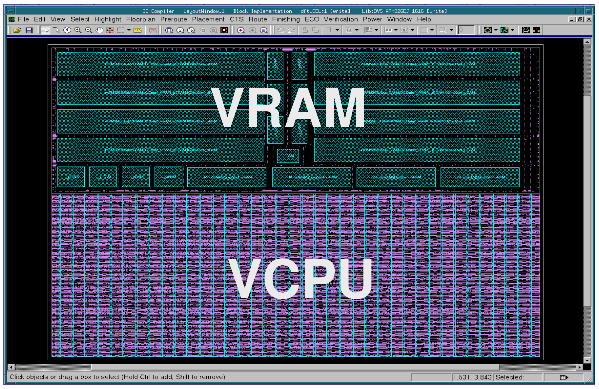


Figure 9 - ARM926 core floorplan

Rail analysis was performed to verify the IR drop through the VDD mesh and across the power gates. It was found to be 18mV well within the 50mV budget (5% of 1.0V nom supply)

To manage the in-rush current during startup, a dual switch network design was implemented, one providing the “soft start” as a daisy chain of weak “starter” power gates and another, “main” network of full power gates which turned on when the virtual rail reached a predetermined level.

The control logic that was built around a Schmitt trigger sensed the level of the switched “virtual” V_{DD} around 90% of desired voltage and subsequently turned on the main network.

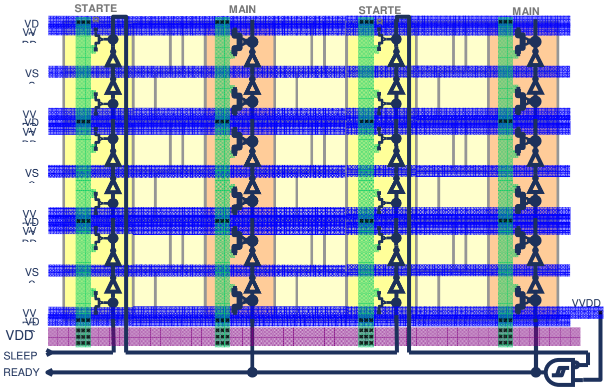


Figure 9 - MTCMOS switch control network

On running post layout circuit simulation to verify the in-rush current and switch on times, the peak in-rush current was no more than 80mA and it took just under 100nS from de-asserting logical SLEEP to bring the switched “virtual” V_{DD} up to operating voltage and for the Schmitt trigger to fire and assert READY(Figure 10).

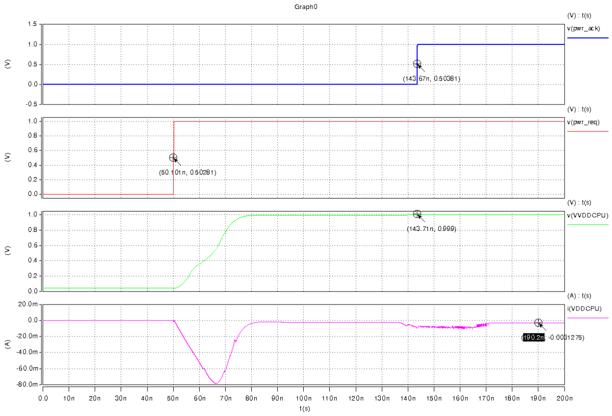


Figure 10 – MTCMOS control network power up simulation

The de-assertion of the control signal to the power gates disconnects the power from both the ARM926EJS processor and the OTG USB core, cuts down leakage aggressively when in the light sleep mode. However to ensure a quick restoration of state when the design is brought out from the power gated mode back in to active mode it is necessary to retain the original state of the sequential elements.

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

The advantage of retention registers is the simplicity and efficiency in save and restore of overall design state. They have a relatively low energy cost of entering and leaving standby mode and so are often used to implement “light sleep”. In order to minimize the leakage power of these retention registers during power gating it is important that the storage node and associated control signal buffering is implemented using high threshold low leakage transistors.

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

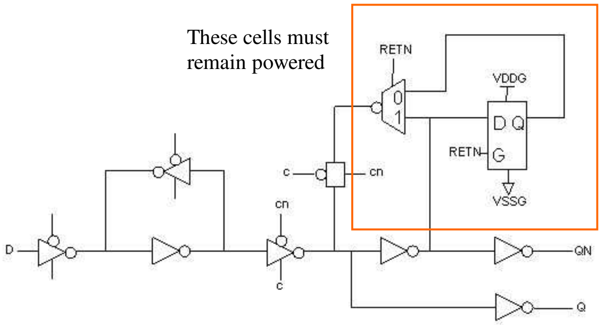


Figure 11 - PMK Retention Register

3.4 Scan Based Hibernate to Enable Deep Sleep Mode

If the system is in standby for a sufficiently large time, it is possible to store the state in main memory and cut the power to all logic including the retention registers. It has a higher energy cost during transience i.e. entering and leaving standby mode but makes up for that by not leaking at all during the off state and is used to implement “deep sleep” mode.

To facilitate this, a novel bus transaction based technique was developed to save and restore state to any AHB connected memory. It made use of overloading the scan logic built in as part of DFT. It utilized the existing scan structure to shift out the current state of the registers and save it externally into the memory in the SoC.

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 32 bit words to a dedicated AMBA bus master to be saved to memory (Figure 12). The design of this dedicated bus master included an implementation of the “CRC-32” algorithm to check the integrity of the restored the data.

Besides using this to scan out the state of the design, this “Scan Hibernate” system can verify the integrity of the state restored from the retention registers.

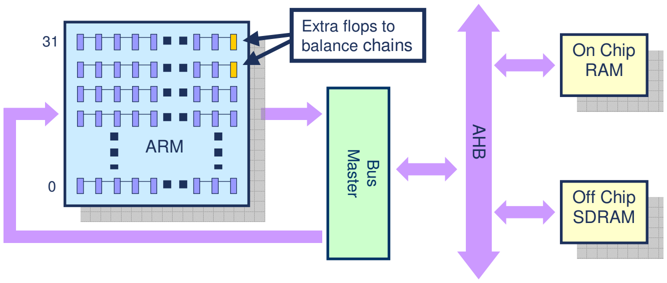
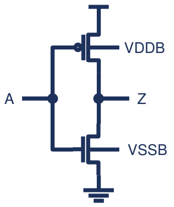


Figure 12 - Scan Hibernate

This can be done by storing the state to memory as well as the retention registers before entering standby 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

3.5 VTCMOS



The ARM926 core also made use of Variable Threshold CMOS (VTCMOS) is another very effective way of mitigating standby leakage power.

This required a triple well process so that “deep” n-wells could be placed under the p-wells in order to isolate them so that they can be held at different potentials. This required a “tapless” library with floating wells so that special tap cells which have independent contact with the wells could be placed at regular intervals to set the body bias. These special well bias cells then needed to be connected together with two power meshes, one for the nwell and one for the pwell.

Since 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 in to the layout of each power gate cell. The implementation of VTCMOS almost came with no overhead as all the vertical connectivity was done by abutment between each power gate cell just like the SLEEP signal. However the VTCMOS implementation required the placement of special deep nwell "capping" cells on the ends of each standard cell row in order to meet the minimum nwell overlap of deep nwell as prescribed by the process rules.

4 Analysis from SALT silicon

After the SALT chip bring up, as part of the silicon validation, MPEG tests were run on a short movie that ran in an endless loop in a 25 fps workload. Besides that, Dhrystone tests were performed to observe the energy consumption at different voltage, frequency points.

Table 1 – SALT VDDCPU supply current measurements

Current (mA) - CPU (std cell)

Voltage:	1.10	1.00	0.90	0.80	0.70
300MHz	52	45	39	33	X
200MHz	37	32	27	23	20
100MHz	22	18	15	13	11
CG	6	5	4	3	2
SRPG	<1	<1	<1	<1	<1

Figure 13, below, shows power measurements done on the ARM926 CPU in the SALT chip, performed at room temperature (22C).

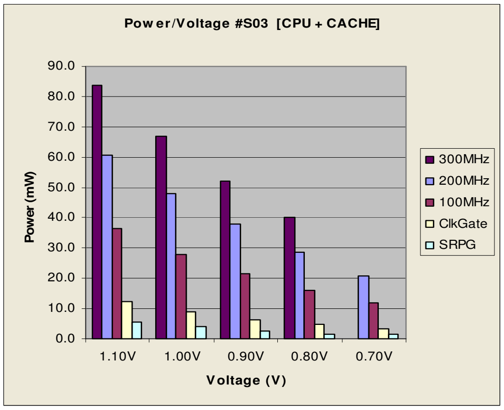


Figure 13 - Measured Power at different modes

At 1.0V, it was observed that the CPU core dissipated 0.133mW/MHz + 5mW of leakage and the CPU RAM dissipated 0.06mW/MHz + 4mW of leakage power. Total power consumption was 67mW @ 300 MHz dynamic power and 9mW of leakage.

In the light sleep mode, when the CPU core was power gated and state stored in the retention registers, the leakage power was cut by over 96% (observed at 140uA drawn at 1.0V). On the RAMs, reducing the ram supply voltage to 75% of nominal gave close to 40% reduction in RAM leakage.

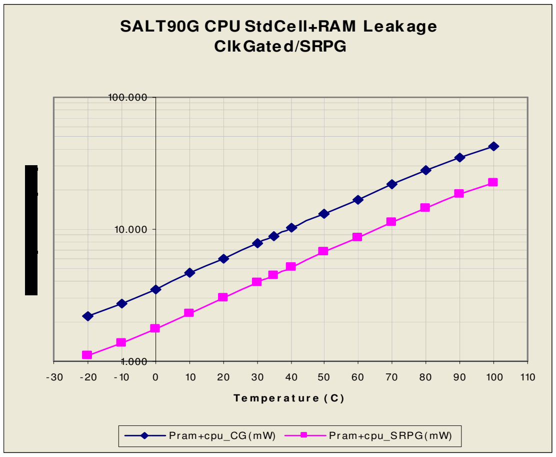


Figure 14 - SALT chip SRPG vs HALT leakage with temperature

The above figure shows the ARM926 leakage over temperature (log scale). The "blue" (darker) graph represents the leakage in the Halt mode where the system is halted but not power gated. The pink (lighter) graph represents the leakage in the Light Sleep mode, where the core has been power gated. In both cases the RAM is still active and held at the nominal voltage (1.0V).

It is observed that the overall saving is close to 50% quite consistently over the entire range reducing slightly at temperature closer to 110C as the power gates which determine the leakage in the core tend to have a different characteristic as opposed to the general CMOS logic.

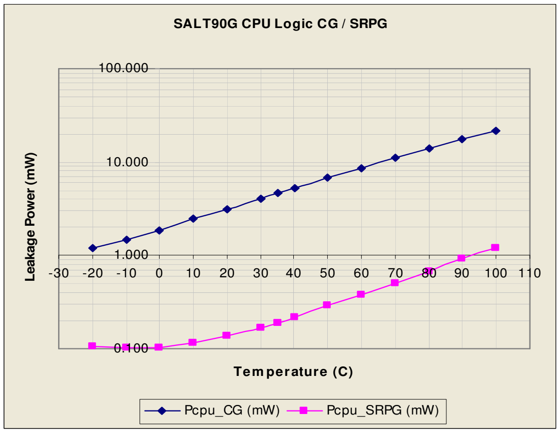


Figure 15 - ARM926 core SRPG vs HALT leakage with temperature

The figure 15 shows comparison for SRPG (Light Sleep) mode with the baseline (Halt mode) for just the CPU core. In this case, the savings are greater than 10x. The increase in CPU SRPG power with temperature is caused because of the increased leakage through the power gates with temperature.
The graph below shows the gain in leakage power savings in the SRPG mode comparing the CPU vs CPU+RAM, this indicates a 20x saving over the portable product range (0-70C)

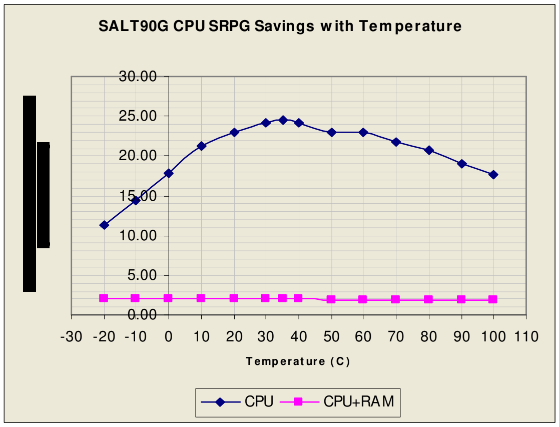
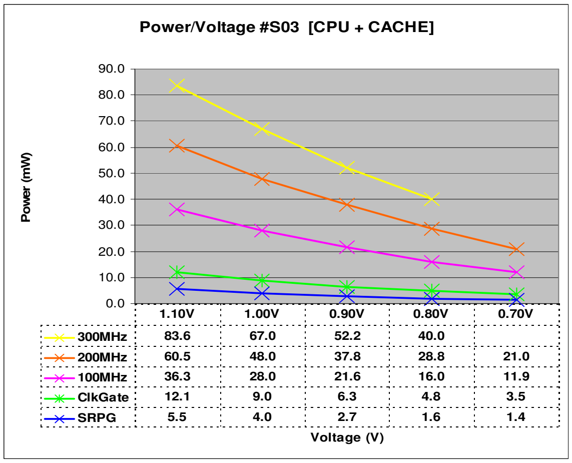


Figure 16 - ARM926 core normalized leakage savings

Figure 17 shows the overall power savings when, for some of the working dice, voltage scaled down to 0.8 V and was able to run at desired performance. The comparison of overall power in runtime vs the gain when the core is in SRPG is also shown. What stands out in the graph is that as the voltage is reduced, the leakage in the Halt mode reduces significantly and the ratio between SRPG mode and Halt mode savings drops as well.



	1.10V	1.00V	0.90V	0.80V	0.70V
300MHz	83.6	67.0	52.2	40.0	
200MHz	60.5	48.0	37.8	28.8	21.0
100MHz	36.3	28.0	21.6	16.0	11.9
ClkGate	12.1	9.0	6.3	4.8	3.5
SRPG	5.5	4.0	2.7	1.6	1.4

Figure 17 – SRPG vs HALT mode leakage savings with reduced voltage

The SALT chip also provided with configuration and debug ports to observe internal nodes to understand different aspects of the power management modes in runtime, when actual workloads were executed on the processor.

In the power management mode, the processor executes an instruction to halt the pipeline and indicates to the control logic (part of the SoC) that it can be put into a desired leakage management mode. The controller can then place the logic in one of the power management modes which are then exited when an interrupt is detected.

Figures 18 and 19 show the latency between an interrupt arrival and the time it takes to get the logic running (shown by de-assertion of the interrupt, when it is serviced), as observed on a digital storage oscilloscope.

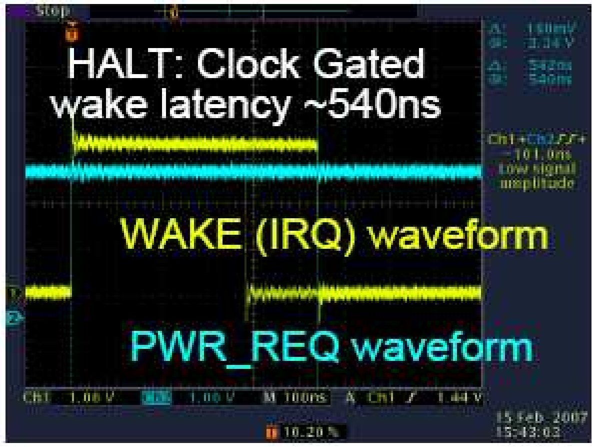


Figure 18 - Architectural Clock Gating wake up latency

The wakeup latency difference between the 2 modes is about 190ns.

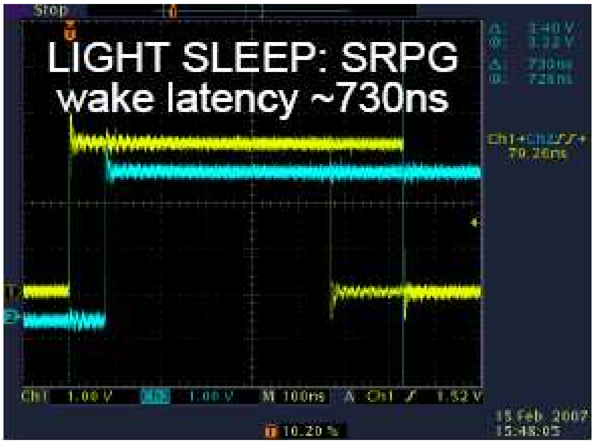


Figure 19 – Light Sleep wake up latency

Figure 20 shows the latency associated with bringing up the CPU state using the scan based restore method in which the register state that was stored in the system RAM is shifted back into the ARM926 registers.

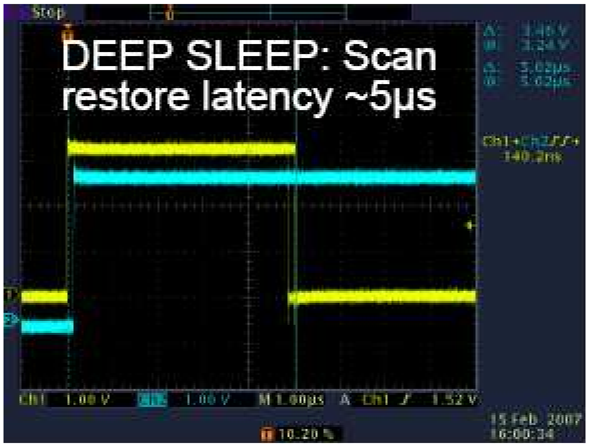


Figure 20 - Deep sleep wake up latency

Figure 21 is the diagnostic mode run on the ARM926 to check the integrity of the the retention registers when in the SRPG mode. The steps involve running a CRC-32 check on the entire packet that comprises the state of the CPU and checking that against the residue that was stored in the main memory prior to the power gating.

Over a set of chips and the tests done so far, we have been able to show that the retention elements save state reliably over a broad temperature range between the operating voltage range of 0.7-1.1V.

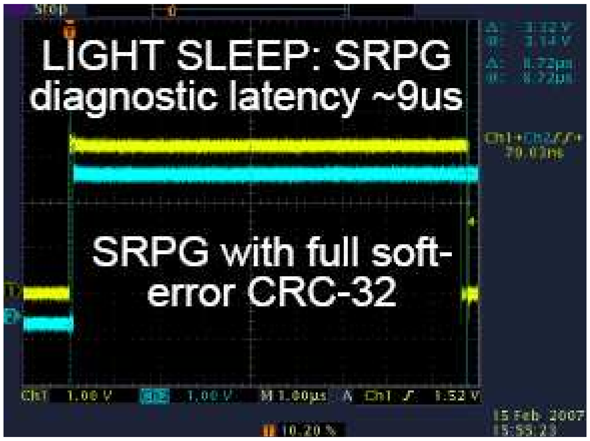


Figure 21 – Scan Hibernate used in diagnostic mode

The increase in temperature has an interesting effect on the restore latency which has been shown in figure 22. As the leakage through the switches increases, the virtual VDD rail which is driven by the power gates does not discharge completely, but the "settled" potential on that node tends to increase with temperature. This results in a faster response during the restore as shown in the figure below. The sweep shows the latency smear as the window formed starting with the request to turn on the switches (PWR_REQ) to power acknowledge signal (PWR_ACK) starts to shrink as the ambient temperature increases.

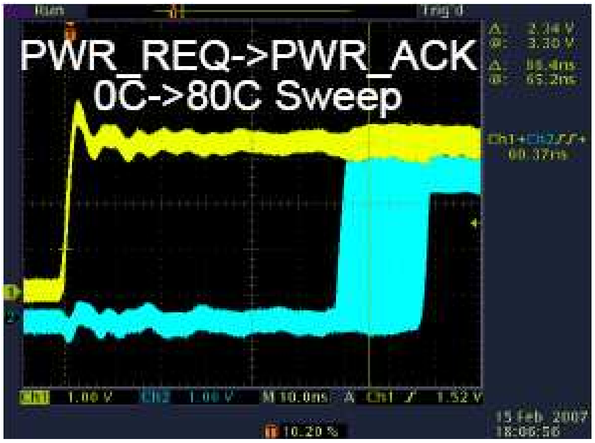


Figure 22 – Sleep to Active latency plot with temperature sweep

5. Test Challenges in SALT design

Although the ARM926 core was constructed of 32 scan chains to facilitate the Hiber scan on a word boundary, at the the chip level, the SALT design was made to operate with 8 scan chains.

To ensure that the MTCMOS switches were controlled during shift and capture, we added DFT fixes to ensure that the control logic in the ILC (energy/leakage controller block) could be designed to get a high coverage. At the same time, the MTCMOS control signals were held to fixed value to facilitate ATPG for the ARM926 core.

To test the logic on the switch network that formed the daisy chain we could control and observe the PWR_REQ (signal that turns on the weak header network) as well as the signal that became active once the entire set of 1600 switches (weak and strong) were turned on. This provided limited coverage of the switch network helping us understand if there were issues with the repeaters on the MTCMOS control network. However this method did not allow us to control the MTCMOS switches directly. To facilitate the control and observation of the MTCMOS switches, an indirect method was used. This required us to override the analog control block on the switch control network (modeled as a Schmitt Trigger), and allow the control of the network externally from a primary port. By observing the response of the time it took for the Virtual VDD to ramp up, we are able to determine if there is a weakness in the network.

This method however does not provide us finer granularity by which we can control and observe a set of switches to diagnose weakness in a particular region on the die where the switches are weak or let us study wear out effects on a per column basis. This limitation is being addressed in the next version of the design based on ARM1176 that we are designing. .

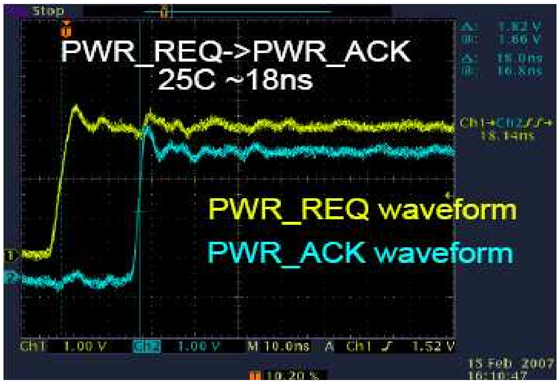


Figure 23 – PWR_REQ(primary input) to PWR_ACK(primary output) to observe MTCMOS switch drive capability

6. Conclusions

With the SALT design, we were able to demonstrate that a general purpose process could be used to get effective leakage reduction as well when the design was in a standby mode. However the implementation of the design came with several challenges associated with power gating such as managing in-rush current during transience, ensuring that the power gates operated in a linear mode with minimal drop during peak performance as well as methodology challenges such as integration with existing standard ASIC flows and testability of the power gating network.

Much of silicon validation on the SALT chip continues as this paper is written. We are modulating the VDDP and VDDN voltages that control source to substrate bias of the ARM926 transistors to manage performance and leakage during runtime as well as aggressive back biasing to manage leakage in standby.

The other aspect that we are planning to test is the wear out of the power gates by accelerated aging and measure the impact on reliability as well as effects of NBTI. To address deficiencies with testability of the power gating network, in the next version of the design we have built BIST logic that can detect weakness in the power network because of faults in the power gates.

7. Acknowledgements

I would like to acknowledge the contribution of my colleagues John Biggs, Alan Gibbons, David Flynn, David Howard, Mike Keating and Sachin Rai. The joint effort on part of designers from ARM and Synopsys was critical to get silicon that worked successfully as per the specifications.

8. References

[1] M. Powell, S.-H Yang, et. al. “Gated-Vdd: A circuit technique to reduce leakage in deep-submicron cache memories”, in Proc. Int. Symp. Low Power Electronics Design, pp. 90-95, 2000.

[2] D. Flynn, K. Flautner, D. Roberts, D. Patel, “IEM926: An Energy Efficient SoC with Dynamic Voltage Scaling”, Proc. DATE, pp. 324-327, 2004.

[3] S. Borkar, “Design Challenges of Technology Scaling” *IEEE Micro*, pp. 23-29, Vol. 19, Issue 4, 1999.

[4] B. Calhoun, F.A. Honore, & A. Chandrakasan, “Design Methodology for Fine-Grained Leakage Control in MTCMOS”, Proc. Int. Symp. Low Power Electronics Design, pp. 104-109, 2003.

[5] K. Roy, S. Mukhopadhyay, H. Mahmoodi-Meimand, ”Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits”, Proc. IEEE, pp 305-327, Vol 91, No. 2, 2003.

[6] M. Keating, D. Flynn, R. Aitken, A. Gibbons, K. Shi, *Low Power Methodology Manual*, Springer 2007.