Low power reduction techniques for Ultra Low Power Processors

Karthik Sukumar
Department of Electrical and Computer Engineering
Technical University Munich (TUM)
Munich, Germany
karthik.sukumar@tum.de

***Abstract*—In this paper the works of J.Zhou et al. [1] and J.Tang et al. [2] on design techniques for an ultra low-power processor is reported.**

J.Zhou et al. focusses on Near-Threshold processor design for Ultra low power processors. The paper discusses the design challenges, performance issues and solutions of the above. The circuit level power reduction techniques are addressed here.

J.Tang et al. discusses a case study by using GPS to demonstate that an Ultra-low power processor used along with a heavy applications processor can lead to further power savings. The architectural level power reduction techniques are addressed here.

***Index Terms*—Near-Threshold, Ultra low power, GPS, Offload Co-processor, Interrupt service routines**

I. Introduction

Power constraint portable devices (e.g. Mobile phones, wearables, IoT devices) require ultra low power processors in order to save battery life. Such low power devices enable operation with the help of energy harvesters or small baterries. Sub and Near-Threshold devices have been shown to have an overall low power consumption and past works confirm that Near-Threshold designs are more efficient than using clock and power gating.

For further reduction of power consumption without loss of performance we need to look at architectural level improvements to preseve scalabililty.

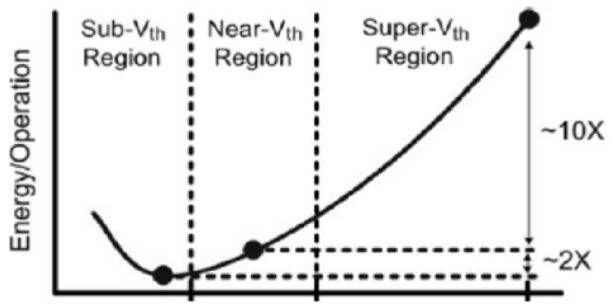

Fig. 1: Sub/Near-Threshold operation. Source: [1]

Near-Threshold designs refer to operating the transistor slightly above the threshold region. The major advantage of Near-Threshold over Super-Threshold operation is the decrese in energy per operation. Dynamic energy per operation increases quadratically with voltage as ($P_{dyn} = fC_L V_{DD}^2$), where as static energy per operation increases with decreasing voltage. This is because the static power is constant and the static energy consumption increases with increasing time period per operation (As max frequency of operation reduces with decresing supply voltage). Hence by adding the two we get a relationship as shown in Fig. 1. Although we see that the Sub-Threshold operation has lower energy than the Near-Threshold, we want to find a balance between performance and energy per operation and hence we will investigate the functionality, variability issues and performance challenges of Near-Threshold design in this paper.

II. Functional Challenges of Near-Threshold Design

Past research has shown that Near-Threshold design is easy to implement for ALUs and logic gates. But SRAM and level Shifter have their challenges in operating in the Near-Threshold region.

A. Level Shifters

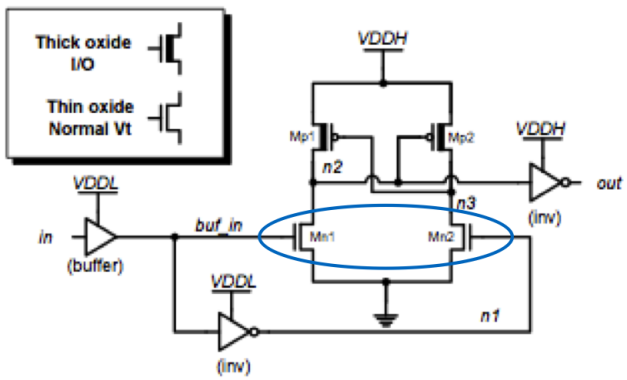

Fig. 2: Weak Pull ups Source: [3]

In the Near-Threshold operation the pull down network as shown in the Fig. 2 is weaker than the pull up network and consequently the the level shifter cant pull down to ground.

The issue of converting a Near-Threshold voltage to a super-threshold voltage therefore needs complex circuitry which is

not only low power in nature but also scales in performance with voltage.

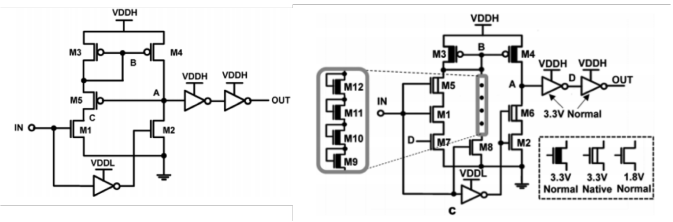

Fig. 3: Weak Pull ups Source: [6]

In [4] [5] [6], current mirrors based design is suggested as it provides performance scalablity and high energy efficiency as shown in Fig. 3.

B. SRAM Design Challenges

There are 2 issue with Near-Threshold SRAM design. One is that many cells share the read/write interface and the second being that the SRAM prefers small transistors for increased density. These conventions make it difficult for them to function in the Near-threshold region. Small transistors have small current driving strength and large variation at low voltages, causing read/write disturbances.

The solution to both the issues is to have a decoupled read write interface namely an 8T/10T SRAM design. Although not as dense as the traditional 6T design, they are more resistant to manufacturing variabilities.

III. Performance Improvement for Near-Threshold Devices

A. Near-Threshold Device Sizing

It has been found that the parasitic effects dominate more in the Near-Threshold than the Super-Threshold regions. Reverse Short Channel Effects (RSCE) for example is rather strong in the Near-Threshold region. As opposed to DIBL(Drain Induced Barrier Lowering) the threshold voltage decreases with increasing Channel length. Another effect is INWE(Inverse Narrow Width Effect) which causes the threshold voltage to decrease as the transistor width decreases. Minimum finger widths are therefore used to reduce delays and improve energy efficiency. By making use of the minimun-sizing method and by selectively tuning the length of critical and non-critical paths the energy efficiency can be improved while meeting the performance specification.

B. Wide Dynamic Voltage Scaling

Designing with a large voltage swing from Near/Sub-Threshold voltages to Super-Threshold voltages (e.g. from 280mV to 1.1V) enables the processors to go to deep sleep mode when not active and spull up to full power when active. This fast voltage switching is critical for energy efficiency without loss of performance.

C. Parallel Processing

The performance degradation of the Near-Threshold operation can be compensated by parallel processing. In [8], 4 homogeneous parallel processing engines have neen designed to accelerate Near-Threshold JPEG encoding, achieving 40MB/s (or VGA, 30 fps) and a 1.33pJ/cycle at 0.6V in a 65nm process.

IV. Manufacturing Variabilty Challenges

Manfacturing variabilty plays vital part as it leads to large performance variations. As the current has an exponential relationship to voltage around the Threshold voltage, small variation in voltage can result in a large current. Therfore process variation can cause a significant variablity in delay or performance. Some of the ways mentioned below can mitigate such risks.

A. Library Pruning

Its important to characterise the standard cells at low voltage, this way the one with large performance variations and poor performance are taken out e.g: Gates with 4 stacked transistors.

B. Pipeline Optimization

Optimising pipeline depth has shown to decrease processor variabilty. It has been found that large logic depths reduce variabilties through the averaging effect. Therefore shallow pipelines with larger logic depth are adopted as a measure against process variations. As shown in Fig. 4 the 2-stage pipeline with 75 logic gates per stage reduces the standard deviation by 19% compared to design with 10 gates per pipeline stage.

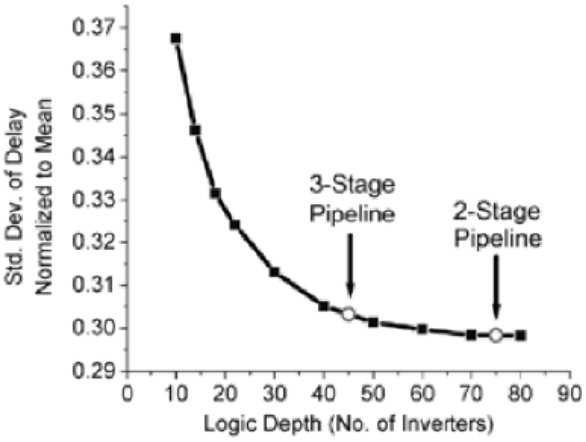

Fig. 4: Random variation vs number of pipeline stages [1]

V. Architectural Level Power Reduction Techniques

For high performance embedded devices we need to consider architectural design changes to maintain the performance. The Application Processor(AP) is the most power consuming

component in the design. In order to reduce the power consumption as much as possible we need to wake up the AP only when its absolutely necessary. Given that the peak workload on most mobile embedded devices is often periodic interrupts or user requests we can utilise this to offload the AP of servicing these routines.

This is illustrated with the help of a case study to show the advantages of using an Ultra Low Power Offload Co-processor(OCP), which could be one of the processors incorporating all the circuit level design techniques described in the earlier sections.

A. Case Study: GPS

GPS is chosen for our case study as it usually samples at 1Hz and is a periodic event. Research has shown that GPS can consume upto 95% of the total power consumption of the mobile device [7].

The system we are considering here is a typical ARM11 micro-processor which is commonly found in most mobile phones these days. In the power profile for the AP shown in Fig. 5 the CPU is initially in sleep mode consuming 20mW before the interrupt occurs. The interrupt is serviced in 2ms (at 1000mW) and the AP dosent transition to sleep mode immediately, instead it must operate at a lower frequency consuming 650mW for 100ms. It drops into yet another mode at 400mW for another 100ms. Only after executing in all these intermediate modes does the CPU go back into deep sleep. Therefore although the interrupt only takes 2ms to execute the application processor remains active for over 200ms consuming 60mJ of energy.

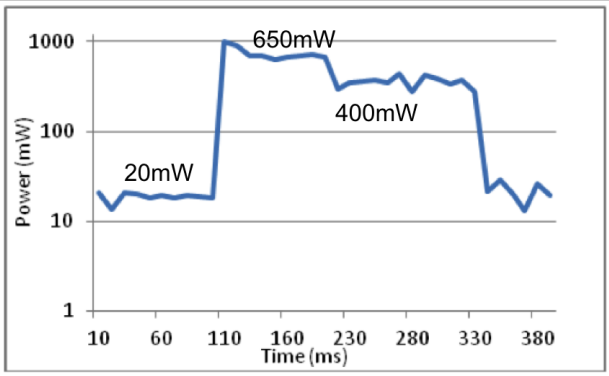

Fig. 5: Power Profile when a GPS interrupt occurs for an ARM11 CPU: [2]

VI. OCP Design

An alternative SoC design with a OCP is shown in Fig. 6. In our case study the Application Processor is an ARM11 CPU and the OCP is a MIPS32 4KS.

A. Energy Analysis

We assume a 1500mAh battery at 3.7V. The values of different modes of both the processors are shown in Table. I. The $P_{AP-active}$ is derived by 60mJ/200ms, which is 300mW.

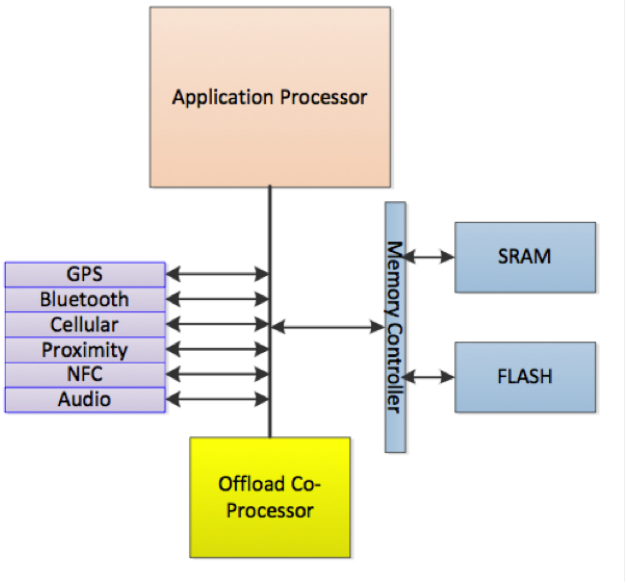

Fig. 6: SoC Design with Offload Co-Processor(OCP): [2]

TABLE I: Power Consumption of different modes

Item	Power(mW)
$P_{AP-sleep}$	20
$P_{AP-active}$	300
$P_{OCP-sleep}$	1
$P_{OCP-active}$	60

$$E = P_{AP-sleep}(1 - A_1)T_1 + P_{AP-active}A_1T_1 \quad (1)$$

The above Equation. 1 holds for the case of using just the AP. We found out through measurements that the A_1=20% and T_1=200ms. The battery would last for a total of 73 hours under these conditions.

$$E = P_{AP-sleep}T_2 + P_{OCP-sleep}(1 - A_2)T_2 + P_{OCP-active}A_2T_2 \quad (2)$$

The Equation. 2 holds for the case of using the AP along with the OCP. We found out through measurements that the A_2=1.5% and T_2=15ms. The battery would last for a total of 254 hours under these conditions

Using Equations 1 and 2 we infer that the we can get a battery life improvement of 3.5 times compared to the baseline.

B. Considerations for OCP Design

1) Different ISA: For the OCP to service the interrupts of the Application processor, the Interrupt Service Routine (ISR) must be a subset of the application processor, such that if the OCP cant handle the interrupt it can just be passed on to the application processor.

2) Interrupt Overheads: The OCP should have a short interrupt handler, where the context switch occurs and the registers need to be saved on the stack. The transition can be performed between 20-30 cycles of overhead for an OCP.

3) Interrupt Overflow: In the case that the interrupt can´t be handled by the OCP, it needs to forward the interrupt or user request back to the application processor.

4) Hardware Acceleration: Interrupt overheads can be minimised by using hardware accelerators based context switches and interrupt overflows handling.

VII. Conclusion

Circuit design techniques are required to tackle the major challenges of the Near-Threshold processor design. Architectural design techniques are required to add additional power and energy saving. We have demonstrated from the OCP based SoC design that the battery life can be extended by 3.5 times by adapting the hybrid approach.

References

[1] J.Zhou, T.H.Kim and Y.Lian, "Near-threshold processor design techniques for power-constrained computing devices," 2017 IEEE 12th International Conference on ASIC (ASICON), Guiyang, China, 2017, pp.920-923

[2] J. Tang, C. Liu, Y. L. Chou and S. Liu, "OCP: Offload Co-Processor for energy efficiency in embedded mobile systems," 2013 IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors, Washington, DC, 2013, pp. 107-110

[3] Y. S. Lin, et al., "Single stage static level shifter designfor subthreshold to I/O voltage conversion,in Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED), Aug. 1113, 2008, pp.197200

[4] S. Lutkemeier, et al., "A subthreshold to above-threshold level shifter comprising a wilson current mirror," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 57, no. 9, pp. 721724, Sep. 2010

[5] Y. Osaki, et al., "A low-power level shifter with logic error correction for extremely low-voltage digital CMOS LSIs," IEEE J. Solid-State Circuits, vol. 47, no. 7, pp.17761783, Jul. 2012

[6] J. Zhou, et al., "An Ultra-Low Voltage Level Shifter Using Revised Wilson Current Mirror for Fast and Energy-Efficient Wide-Range Voltage Conversion from Sub-Threshold to I/O Voltage," Circuits and Systems I: Regular Papers, IEEE Transactions on , vol.62, no.3, pp.697,706, March 2015

[7] N. Priyantha, D. Lymberopoulos, J. Liu, "EERS: Energy Efficient Responsive Sleeping on Mobile Phones, in Proceedsings of the International Workshop on Sensing for App Phones", November, 2010

[8] Y. Pu, J. Pineda de Gyvez, H. Corporaal and Y. Ha, "An Ultra-Low-Energy Multi-Standard JPEG Co-Processor in 65 nm CMOS With Sub/Near Threshold Supply Voltage," IEEE Journal of Solid-State Circuits, vol. 45, no. 3, pp. 668-680, March 2010