2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems

Energy Efficient Memory Decoder Design for Ultra-Low Voltage Systems

K R Viveka and Bharadwaj Amrutur
Electrical Communication Engineering Department, Indian Institute of Science, Bangalore.
Email: {krviveka, amrutur}@ece.iisc.ernet.in

***Abstract*—This paper presents a low energy memory decoder architecture for ultra-low-voltage systems containing multiple voltage domains. Due to limitations in scalability of memory supply voltages, these systems typically contain a core operating at subthreshold voltages and memories operating at a higher voltage. This difference in voltage provides a timing slack on the memory path as the core supply is scaled. The paper analyzes the feasibility and trade-offs in utilizing this timing slack to operate a greater section of memory decoder circuitry at the lower supply. A 256x16-bit SRAM interface has been designed in UMC 65nm low-leakage process to evaluate the above technique with the core and memory operating at 280 mV and 500 mV respectively. The technique provides a reduction of up to 20% in energy/cycle of the row decoder without any penalty in area and system-delay.**

***Index Terms*—Ultra low power, memory interface design, subthreshold, level shifter**

I. Introduction

Ultra-low power (ULP) circuits and systems have gained extensive attention in recent literature, especially for energy constrained applications such as bio-medical implants and autonomous sensors [1]. These operate at very low power/energy levels as battery replacement is very expensive. An effective way to achieve this power reduction is by employing voltage scaling. ULP design thus translates to ultra-low voltage (ULV) design where the supply is scaled to subthreshold voltages.

While conventional CMOS logic circuits have been demonstrated to function down to 180 mV and simple variations of logic style allow operation down to 62 mV, memories have not scaled proportionately. Although SRAMs that function down to 200 mV have been reported [2], memories in general tend to be operated at higher supply voltages compared to logic circuits [3].

Figure 1 shows a typical system, similar to implementations reported in [3] and [1], highlighting the memory interface section of the design. It may be observed here that level shifters are used to interface the core, operating at a lower supply, with the memory that operates at a higher supply voltage. These implementations place the level shifters before the flip-flop (FF) present at the memory interface as shown in the figure. However, memory macros contain logic circuitry such as row decoders that can potentially be operated at lower supplies similar to the core logic. Only the SRAM cells in the memory macro require higher supply voltages to operate reliably.

This paper evaluates an alternate memory interface architecture that enables lower energy/cycle by moving the level shifter into the memory macro. Although level shifters are commonly placed next to the SRAM array [4], this paper evaluates the feasibility, trade-offs and applicability of placing level shifters at various stages along the decoder for ULV systems.

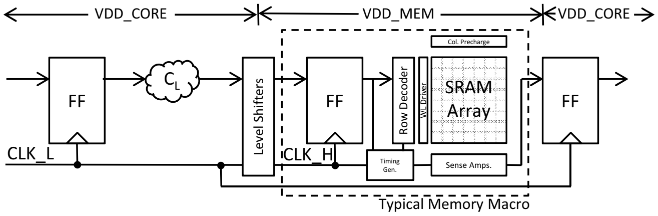
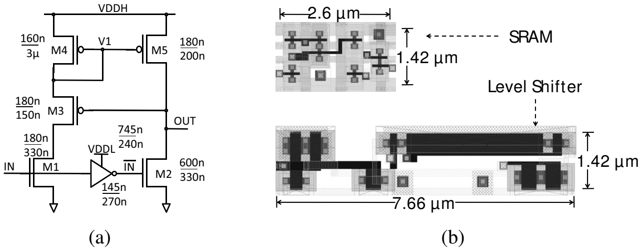


Fig. 1. Generic memory interface of a multi-voltage domain system with level shifters placed before the memory macro.



(a) (b)

Fig. 2. (a) Wilson current mirror based subthreshold level shifter [5]. (b) Layout of 8T SRAM and level shifter of equal pitch.

The rest of this paper is organized as follows. Section II explains the subthreshold level shifter used in our implementation, which is followed by a description of the memory interface architecture in section III. Section IV, then presents the various row decoder architectural design options. Simulation results are presented in section V, we then conclude in section VI.

II. Subthreshold to Above Threshold Level Shifter

Several level shifters capable of translating subthreshold voltages to nominal level have been proposed in literature [6] [5]. The Wilson current mirror based design proposed in [5] employs a technique that lowers the contention between the NMOS pull-down path and the PMOS pull-up path present in conventional level shifters making it suitable for ULV designs.

The level shifters presented in [6] and [5] were designed, and the Wilson current mirror based design [5] (Fig. 2) was

1063-9667/14 $31.00 © 2014 IEEE
DOI 10.1109/VLSID.2014.32

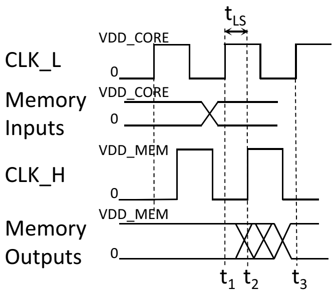


Fig. 3. Timing diagram of the memory interface shown in Fig 1.

chosen, as simulation results showed it to be superior to that of Wooter's design [6] in all performance metrics; delay, leakage power and energy per transition. The design supports a wide range of supplies with $VDDL_{min} = 100mV$ and $VDDH_{max} = 1.2V$, and the case when $VDDL > VDDH$ (with $VDDL_{max} = 1.2V$ and $VDDH_{min} = 300mV$).

III. Memory Interface Architecture

Figure 1 shows the typical memory interface in modern SoCs. A memory control unit (VDD_CORE) generates the inputs required by the memory such as address, chip-select, read/write enable, and write-data, and reads back the data returned from the memory. The memory block is typically available as a macro and is operated at a higher voltage (VDD_MEM). Figure 3 shows the timing diagram of this system. The system clock (CLK_L) is given to the memory after level-shifting (CLK_H), which causes the memory inputs to be latched with a delay equal to the level shifter delay (t_{LS}) at time t_2 as against t_1. However, the memory output is latched at t_3 using CLK_L. Hence the cycle time, T_{cycle}, for this system is given by:

$$T_{cycle} = max(t_{cq} + t_{C_L} + t_{setup}, t_{MEM} + t_{LS}) \qquad (1)$$

where t_{cq} represents the Clk-to-Q delay of a flop, t_{C_L} is the delay in the combinational block labeled C_L in Fig. 1 (t_{C_L} represents the critical logic path delay, which need not be in the memory controller block), t_{setup} is the setup time of a flop, and t_{MEM} is used to represent the delay in the entire memory path (including any flop setup and Clk-to-Q delay). As the core supply is scaled, to meet demands of lower power consumption, the delay of each pipeline stage scales differently. The new cycle time is then given by:

$$T'_{cycle} = max(t'_{cq} + t'_{C_L} + t'_{setup}, t_{MEM} + t'_{LS}) \qquad (2)$$

where the $'$ represents the new (increased) delay corresponding to the reduced core supply voltage. Note that the memory delay has not changed as it continues to operate at the higher supply.

Figure 4 shows the variation of level shifter delay and 20 fan-out-four (FO4) inverter delay (typical gates per pipeline stage in processors [1]), as the supply is scaled. It may be seen that the combinational delay increases at a significantly faster rate compared to the level shifter delay as the supply is reduced. Depending on whether critical path was in memory, before scaling the supply, there are two possible design scenarios.

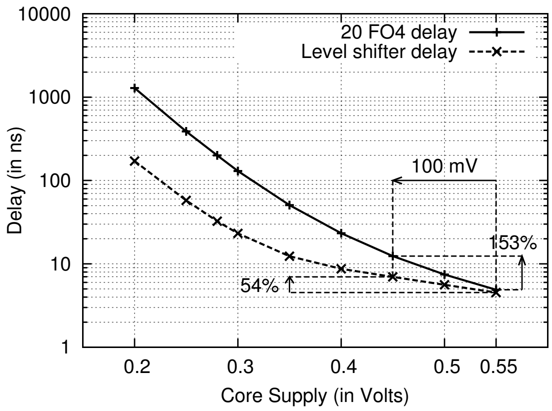


Fig. 4. Variation of FO4 delay and level shifter delay with VDD_CORE.

1) Case 1: If logic path was critical before reducing the supply i.e.

$$T_{cycle} = t_{cq} + t_{C_L} + t_{setup} \qquad (3)$$

This implies that there is already some slack in the memory path and this slack will increase further as the core supply is reduced.

2) Case 2: If the cycle time was limited by the memory before the supply is reduced i.e.

$$T_{cycle} = t_{MEM} + t_{LS} \qquad (4)$$

Depending on the initial slack in logic path, as the supply is scaled there exists a crossover point where the logic path becomes critical and slack develops on the memory path. With the crossover point given by

$$t'_{cq} + t'_{C_L} + t'_{setup} = t_{MEM} + t'_{LS} \qquad (5)$$

Figure 4 shows that even for a 100 mV difference in supplies (core at 0.45 V and memory at 0.55 V) the core delay increases by 153% as compared to level shifter delay, which increases by just 54%. Hence even if the supply is scaled by a small amount and for reasonable amounts of initial slack in logic path, the memory path quickly becomes non-critical.

Thus as the supply is scaled, in either of the two cases, a slack develops in the memory path. This slack may be utilized to operate some sections of the memory at the lower voltage, enabling a reduction in system power. In order to do so the level shifter must be moved into the memory macro. The first step, in doing so, is to move the level shifter beyond the flip-flop. This causes the level shifter delay to be a part of the memory access path. Thus the cycle time for this system is the same as in Eqn. (1). Now that the level shifter has been placed just before the memory (preceding the memory address decoder or row decoder) without affecting the timing, we can

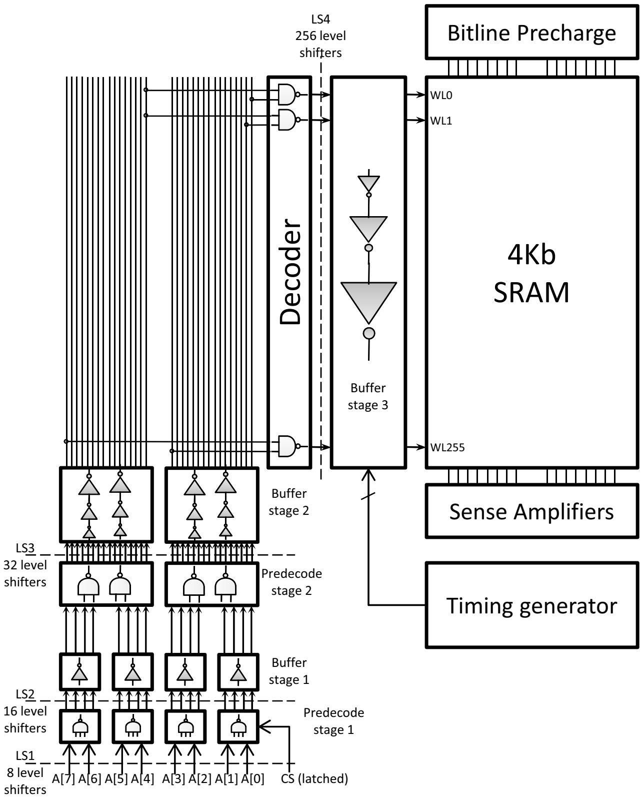


Fig. 5. Proposed Row-Decoder architecture showing various architectural options for placement of level shifters.

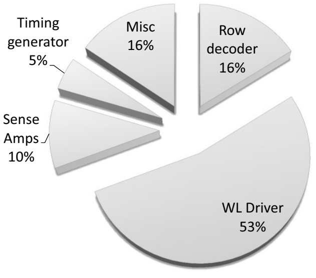


Fig. 6. Typical memory interface leakage power break-up with all sections of the memory operating at 550 mV.

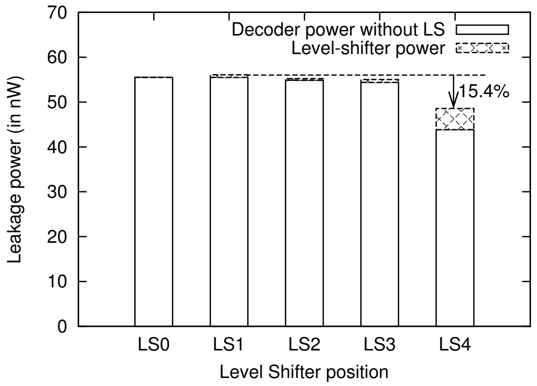


Fig. 7. Decoder leakage power in various level shifter positions.

push it further in as explained in the next section on row decoder design.

IV. Row Decoder Design

The function of the row decoder is to decode the address bits (typically 8-bit as explained in section V) into multiple Word-line (WL) enables, one for each row of the SRAM array. Figure 5 illustrates an 8-bit decoder with multiple stages of pre-decoding. The address bits are decoded in 3 stages using 2 or 3-input AND (NAND + NOT) gates as shown in Fig. 5. All NAND gates are only loaded by 1X (minimum sized) inverters to minimize the effort of the higher fan-in gates (NAND). The outputs of these gates are then buffered to drive their respective load.

The options available for placing the level shifter at various positions in the decoder are also shown in Fig. 5. Mode LS1 represents the case where the level shifters are placed in front of the address decoder (following the flip-flops, as mentioned in the previous section). All blocks of the decoder operate at the higher supply (VDD_MEM) in this mode (table I). This is the supply at which memory runs. The next option would be to place the level shifter at the output of predecode stage 1, denoted as LS2. In this mode the predecode stage 1 blocks would be operating at the lower supply (VDD_CORE) and all other blocks will operate at VDD_MEM. In general all blocks from the input A[7:0] till the level shifters operate at VDD_CORE and the blocks following the level shifter operate at VDD_MEM. The next option is shown as LS3 in the figure where the 32 level shifters are placed at the output of predecode stage 2. The final option is then to place the level shifters just before the word-line drivers (LS4). The number of level shifters required in each mode is also shown in the Fig. 5 and table I. An additional mode, denoted by LS0, is added which is identical to LS1 but with the absence of the level shifters. This mode is used to quantify the penalty incurred by the use of level shifters.

As the level shifter is placed closer to the SRAM WL, more blocks operate at a lower supply voltage. Hence we would expect the delay to increase as we move from mode LS1 towards mode LS4. The energy per transaction, on the other hand, is expected to reduce as more blocks operate at a lower supply and thus consume lower energy. However this trend may be offset by the increase in number of level shifters required as we move from LS1 to LS4. The leakage power will also be affected similarly by the above factors. Another interesting factor, adding to this, is the number of level shifters switching, in each mode, for a given number of address-bits transitioning. As the level shifters are moved closer to the word-line, fewer of them switch for a given number of address bit transitions. However, moving the level shifters closer to the word-lines also causes an increase in area. The results of these trade-offs are studied in next section.

TABLE I
ARCHITECTURAL OPTIONS FOR PLACEMENT OF LEVEL SHIFTERS AT DIFFERENT STAGES ALONG THE ROW DECODER

Mode	Predecode stage 1	Buffer stage 1	Predecode stage 2	Buffer stage 2	Final decoder	Buffer stage 3	No. of level shifters
LS0	High	High	High	High	High	High	0
LS1	High	High	High	High	High	High	8
LS2	Low	High	High	High	High	High	16
LS3	Low	Low	Low	High	High	High	32
LS4	Low	Low	Low	Low	Low	High	256

High – indicates that the block operates at the higher voltage (VDD_MEM)
Low – indicates that the block operates at the lower voltage (VDD_CORE)

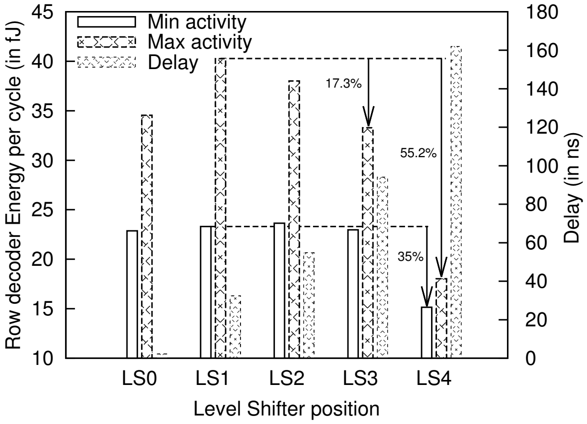


Fig. 8. Decoder Energy/cycle in different level shifter positions for minimum and maximum decoder activity and variation of decoder delay with level shifter position.

V. IMPLEMENTATION AND SIMULATION RESULTS

In order to demonstrate and evaluate the proposed technique, a 4 Kb SRAM (organized as 256 rows by 16 columns) interface has been designed in UMC 65nm low-leakage process. Larger SRAM arrays would generally use column muxing and/or split word-line architecture to address a larger memory space. Hence the analysis presented here is valid even for larger memory sizes reported in [3] and [1].

The SRAM uses an 8T cell (layout in Fig. 2(b)) [7] that contains two transistors for read-buffer in addition to the conventional 6T cell. The memory operates at a fixed voltage of 0.55 V (VDD_MEM) while the core voltage (VDD_CORE) is scaled down to a minimum of 0.2 V, similar to the design presented in [3].

The break-up of leakage power in memory interface circuitry is shown in Fig. 6. The configurations explored in this work only affect the row-decoder power, while the contribution of other blocks remain almost unaffected and act as a static offset to memory power as the modes are varied. We therefore focus only on the row-decoder metrics in this section.

The design presented in [3] operates the memory at 0.55 V and logic, as low as 0.28 V. At these voltages the combinational logic determines the clock frequency to be 5 MHz. Figure 7 shows the variation of the decoder leakage power as the level shifter position is changed under these conditions (with the contribution of the level-shifters and the rest of the decoder shown separately). As the level shifter is moved closer to the WL the total leakage power remains almost constant from mode LS1 through LS3. Mode LS4, on the other hand, provides a 15.4% reduction in leakage power over LS1, thanks to the large buffer stage 2 being operated at the lower supply.

Figure 8 plots the energy/cycle of the row decoder under aforementioned conditions as the level shifter position is varied, for extreme values in activity factor of the decoder. The minimum activity occurs when only one address bit transitions while the worst case activity is observed when all eight address bits transition. Moving the level shifters closer to the WL clearly offers energy benefits with LS4 providing 35% to 55.2% decrease, and LS3 providing up to 17.3%, decrease in energy/cycle over LS1. The figure also plots the increase in delay of the decoder as level shifters are moved closer to the WL. Comparing this with the Fig. 4 (at 0.28 V), shows that the delay increase in combinational logic (195.1 ns) is greater than the delay increase in decoder in mode LS4 (129.7 ns), thus making all modes feasible.

The core voltage may be varied based on system performance requirements which affects the trade-offs in level shifter placement. This was tested by varying VDD_CORE from 550 mV down to 200 mV. For this entire range, it was noted that the increase in delay of decoder (in mode LS4), as supply is reduced, is less than the delay increase in combinational path, thus making LS4 mode feasible over the entire range of voltages.

Figure 9 shows the variation in absolute energy/cycle of the decoder for modes LS1, LS3 and LS4 as VDD_CORE is varied, with VDD_MEM held constant at 0.55 V. The critical path delay is also plotted, in the figure, to show that the system frequency decreases exponentially as the supply is scaled down. Reducing the supply decreases the dynamic energy/cycle while the leakage energy/cycle increases. This implies that there exists an energy optimal point at which the energy/cycle is minimum. This optimum point occurs at a supply of approximately 300 mV for most processors [1] and Fig. 9 shows that this is indeed the case for the decoder as well. Mode LS4 causes both the dynamic and leakage energy/cycle to reduce as more sections of the decoder operate a lower voltage. This results in a reduction in total energy/cycle as seen from the figure.

The savings obtained by moving to the architectures LS3

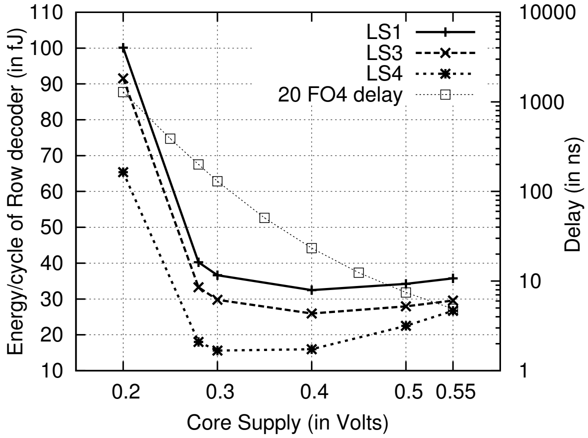


Fig. 9. Variation of absolute Energy/cycle and combinational delay with VDD_CORE.

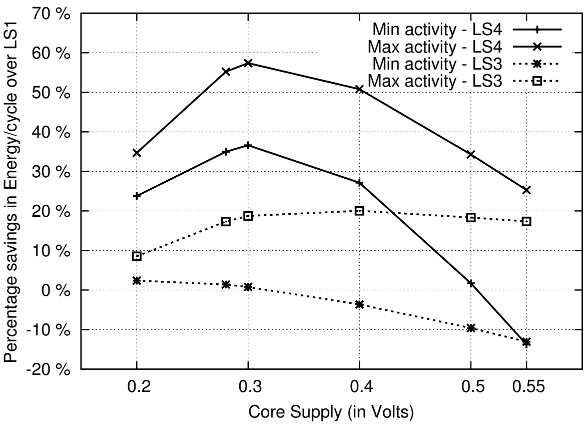


Fig. 10. Percentage saving in Energy/cycle for various values of VDD_CORE, for extreme values of decoder activity.

and LS4 are quantified in Fig. 10. The figure plots variation in percentage savings in energy/cycle of LS4 and LS3 over LS1, as the supply is varied. The results are shown for extreme values of decoder activity. It may be seen that excepting the particular case when both core and memory operate at 0.55 V and only a few address bits transition, mode LS4 always enables reduction of energy/cycle with a maximum savings of 57.4%. The savings are noted to peak in the 300 to 400 mV range (energy optimum VDD) due to the minima in energy/cycle curve in this range.

Mode LS4 requires the level shifters layout to be designed to match the SRAM array pitch, as shown in Fig. 2(b). The 256 level shifters in this mode cause the decoder area to increase by 41%. However in modes LS3, LS2 and LS1 the level shifters are hidden under the long wires at the output of buffer stage 2 avoiding any increase in decoder area. Hence from a practical perspective, LS3 offers a good trade-off with energy savings of up to 20% (Fig. 10) and negligible area overhead.

The minimum energy perspective [1] recommends designing low activity blocks with higher threshold devices to reduce leakage, but running them on a higher supply to maintain performance. As activity in the decoder decreases as one approaches the final WL drivers, LS3 offers a good compromise of separating the higher activity predecoders from the low activity but higher voltage WL drivers.

The only modification required to implement mode LS3, is to shift the buffer stage 2 to make space for the level shifters and separate the power domains appropriately. Thus making this technique amenable to implementation using memory compilers.

VI. Conclusions

The memory interface circuitry for a 256x16-bit SRAM has been designed in UMC 65nm low-leakage process. The core (logic) is operated at supply voltages ranging from nominal down to the subthreshold regime while the memory operates at a fixed voltage of 550 mV. It has been demonstrated, for a wide range of core voltages, that moving the level shifters into the memory macro and placing them close to the word-line drivers enables a reduction in energy/cycle of the row-decoder. This is done by utilizing the slack in memory path obtained by scaling down the core voltage. The technique proposed is shown to be beneficial irrespective of the timing slack present in the core and memory paths before scaling down the core supply. The proposed architecture of pushing the level shifters after the predecoders provides up to 20% reduction in energy/cycle of the row-decoder with negligible area and system-delay overheads.

Acknowledgment

The authors would like to thank Karthik Ramkumar Jeyashankar for valuable discussions, and the Department of Electronics and Information Technology, MCIT, Government of India for funding this research.

References

[1] M. Alioto, “Ultra-low power vlsi circuit design demystified and explained: A tutorial,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 59, no. 1, pp. 3 –29, jan. 2012.

[2] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, “A variation-tolerant sub-200 mv 6-t subthreshold sram,” *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 10, pp. 2338 –2348, oct. 2008.

[3] S. Jain, S. Khare, S. Yada, V. Ambili, P. Salihundam *et al.*, “A 280mv-to-1.2v wide-operating-range IA-32 processor in 32nm CMOS,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, feb. 2012, pp. 66 –68.

[4] K. Brock. (2009) Virage logic: Minimizing design complexity with power-optimized physical IP. [Online]. Available: http://www.powerforward.org/media/p/177.aspx

[5] S. Lütkemeier and U. Rückert, “A subthreshold to above-threshold level shifter comprising a wilson current mirror,” *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 57, no. 9, pp. 721 –724, sept. 2010.

[6] S. Wooters, B. Calhoun, and T. Blalock, “An energy-efficient subthreshold level converter in 130nm CMOS,” *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 57, no. 4, pp. 290 –294, april 2010.

[7] N. Verma and A. Chandrakasan, “A 256 kb 65 nm 8T subthreshold SRAM employing sense-amplifier redundancy,” *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 1, pp. 141 –149, Jan. 2008.