surecore

WHITE PAPER 1

sureCore Low-power SRAM Technology Introduction

TONY STANSFIELD
CTO, SURECORE LIMITED

enquiries@sure-core.com sureCore Ltd, Sheffield Technology Park, Cooper Buildings, Arundel Street, Sheffield S1 2NS, UNITED KINGDOM
www.sure-core.com

SureCore Low-power SRAM Technology Introduction

Tony Stansfield, CTO, sureCore Limited.

Overview

SureCore is a developer of ultra-low power embedded memory for silicon chips. This white paper has been written to provide an overview of the technology developed by sureCore and a comparison against conventional approaches. In developing a technology not driven by V_{min} reduction a number of issues with bit cell stability can be avoided and power consumption can still be dramatically reduced.

Introduction

Modern electronic devices make extensive use of highly integrated electronic components, commonly known as System-on-Chip (or SoC) devices. Static Random Access Memory (SRAM) is a fundamental constituent of all SoCs; Phone SoCs have multi-Mbyte on-chip caches, while in server processors this extends to 10s of Mbytes, and dedicated AI accelerators can have 100s of Mbytes on a single chip. In many cases, SRAM is the single largest component in terms of chip area;

Managing both the active and leakage power consumption in an SoC is critical to extending battery life of handheld devices and to thermal management of high-performance systems. A common methodology used to achieve this is a technique called Dynamic Voltage and Frequency Scaling ("DVFS"), which exploits the fact that active power in a circuit is typically proportional to the operating frequency, and to the square of the supply voltage. DVFS controls both the operating frequency and the voltage to specific areas of circuitry, so that they can be tuned to minimize power whilst maintaining the required level of performance. For example, when part of the SoC is in a standby state all non-essential circuitry can be powered down and the logic running any necessary "housekeeping" functions can have its clock frequency scaled back. This means that vital low level tasks are still running, albeit relatively slowly, and power dissipation is significantly reduced. DVFS is therefore an effective technique for saving power in logic circuits, but is less effective in memory-intensive circuits, due to fundamental limitations on the supply voltage to an SRAM.

Limits to SRAM Supply Voltage

An SRAM has to be able to perform three basic functions:

1. It must be able to retain data.
2. It must be possible to access the data stored in the memory. i.e. it must be readable.
3. It must be possible to modify the data stored in the memory. i.e. it must be writable.

Updated October 2019

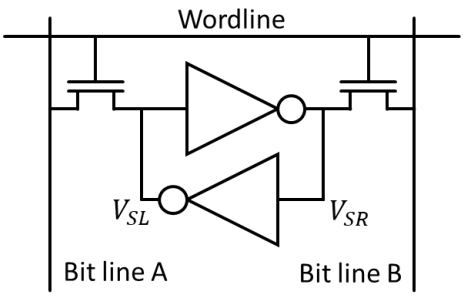

Figure 1 Six transistor SRAM bit cell

Process Variation and Bit Cell Behaviour

All three of these bit cell functions are subject to limitations that are due to the properties of the manufacturing process, and in particular to the variability within the process. The basic unit of an SRAM is the 6-transistor bit cell (Figure 1), consisting of two cross-coupled inverters to store a single bit of data and two access transistors through which the data is read and written. In order to achieve high storage density the bit cell is made as small as possible, which in turn means that the individual transistors are made as small as possible while still achieving reliable bit cell operation. However, small transistors are subject to greater manufacturing variability than larger ones, because of the physical nature of the effects that give rise to manufacturing variability. These include:

- Line Edge Roughness (LER) – the edges of a transistor are not the straight lines that appear in a layout editor, nor the smooth curves suggested by OPC.[1] Instead the edges have a rough appearance due to the granularity of the materials involved. This means that any transistor channel will have some segments longer than the nominal channel length and other segments that are shorter than this. The wider the transistor, the more scope there is for spatial averaging to reduce the effect of process variability, or alternatively the smaller the transistor the greater the potential impact of variation.
- Random Dopant Fluctuation (RDF) – in a transistor with a doped channel the number and positions of dopant atoms is subject to statistical variation. Again, there is scope for averaging across the width of the transistor, and so the variability impact is greater in a smaller transistor.
- Metal Gate Granularity (MGG) – the positions and orientations of grains in the gate material affect gate work function, and therefore affect transistor threshold voltage.

All these effects lead to ***local*** variation in transistor behaviour, meaning that there is little or no correlation between their impact even on physically adjacent transistors. However, they can be simulated in order to estimate the likely range of transistor behaviour, and that spread in likely transistor behaviour can then be included in a Monte Carlo circuit simulation to estimate the impact of process variation on circuit behaviour. Figure 2 shows the result of such a simulation of the two inverters in a 6-transistor bit cell for a hypothetical 20nm bulk CMOS process.

[1] Optical Proximity Correction

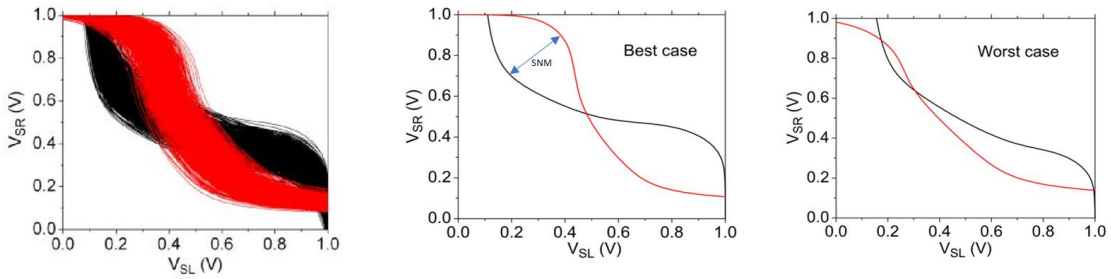

Figure 2 Simulated bit cell inverter characteristics in the presence of variability[2] (a) 1000 cases (b) best case (c) worst case.

The red and black curves in Figure 2 show the characteristics of the two inverters in the bit cell – i.e. the relationships between the nodes V_{SL} and V_{SR} shown in Figure 1. The larger the space between the curves then the harder it is to disturb the state of the cell. Static Noise Margin (SNM) is a measure of the space between the two curves,[3] and therefore is a useful measure of the ability of a bit cell to retain data. Figure 2 (b) shows that the best-case SNM in this simulation is large, and therefore under nominal conditions the cell can retain data. However, the worst-case of the 1000 simulations – Figure 2 (c) – has a greatly reduced SNM, indicating that its data retention is significantly worse than the best case. Figure 2 (a) shows the inverter characteristics from all 1000 simulations superimposed, and so gives an idea of the likely spread with 1 million bit cells. Here there is essentially no gap between black and red ranges, indicating that in a sample of a million bit cells there will be cases that are unable to retain data. Since state-of-the-art SoCs can contain 100s of millions of bit cells, the bit cell simulated in Figure 2 is clearly unsuitable for use in such a device and would require changes to either the process, the transistor sizes in the bit cell, or the operating voltage in order to reduce the impact of variability. Since raising the voltage in order to increase SNM goes against the goal of being able to operate at low voltage in order to save power, in practice it is the process or the transistor sizes that need to change.[4]

Similar Monte Carlo simulation techniques can also be used to assess cell readability (e.g. by simulating the spread of currents that can be drawn through the access transistors, which determines the time taken to discharge a bit line) and cell writability (e.g. by determining the wordline and bitline voltages required to reliably flip the bit cell state)

Process Variation and SRAM Operating Voltage

The preceding section indicates the importance of statistical simulation in predicting SRAM behaviour. Since modern SoCs can contain 10s or 100s of millions of bit cells it is important to understand the behaviour of the bit cell in the extremes of the process distribution, and it is these extremes that determine the limits on SRAM operating voltage. These limits are usually expressed as two key parameters:

- V_{ret} – the minimum voltage at which the bit cell can reliably retain data.
- V_{min} – the minimum supply voltage at which an SRAM can reliably read and write.

---

[2] Source – Gold Standard Simulations Ltd, 2013. Results of simulating 1000 nominally identical bit cells with local variability.

[3] Static Noise Margin is the longest diagonal distance across the smallest space between the red and black lines.

[4] In fact, below 22nm it is the process that is changed – both FinFET and FDSOI processes use an undoped channel in order to remove the Random Dopant Fluctuation component of transistor variability.

V_{ret} is determined by the properties of the bit cell (it is the minimum voltage that has an acceptable worst-case SNM), while V_{min} is set by a combination of bit cell and other SRAM circuits. V_{ret} is an absolute limit on memory operation, and so normally $V_{min} > V_{ret}$.

Since V_{ret} and V_{min} are limits on SRAM supply voltage, they are also lower limits on the voltages that DVFS can apply to an SRAM in order to save power. Unfortunately, these limits are higher than those that can be applied to on-chip logic – as a rule of thumb, V_{ret} is often around two thirds of the nominal voltage for a particular process, and V_{min} may only be 10% below nominal, whereas logic can often operate below 50% of the nominal voltage.

Alternative Ways to Save Active Power in SRAM

V_{ret} and V_{min} set limits on power saving via DVFS in an SRAM, and these limits are worse than those for on-chip logic. SureCore has therefore developed alternative techniques for saving power in SRAMs that do not rely on reducing the supply voltage, and which are therefore not limited in their effectiveness by V_{ret}. These techniques are independent of manufacturing process, and have been successfully demonstrated in bulk CMOS, FDSOI, and FinFET processes.

A Model of SRAM Active Power

Fundamentally, the predominant cause of active power dissipation in an SRAM is the movement of charge on and off the parasitic capacitances within in the SRAM. We can therefore express the energy, E, in a memory access that is due to a component i, as:

$$E_i = n_i \Delta Q_i V_{dd}$$

$$E_i = n_i C_i \Delta V_i V_{dd}$$

Where n_i is the number of instances of the component, ΔQ_i is a change in the charge associated with it, and V_{dd} is the supply voltage that is ultimately the source of the charge. ΔQ_i is itself the product of the relevant capacitance C_i, and the voltage swing on that capacitance, ΔV_i. Total active energy is then the sum of all such terms:

$$E = \sum_i E_i = \sum_i n_i C_i \Delta V_i V_{dd} = V_{dd} \sum_i n_i C_i \Delta V_i$$

In order to achieve power saving in a way that does not rely simply on scaling V_{dd}, it is necessary to reduce the n_i C_i ΔV_i terms in this expression. The various components in an SRAM can be characterised by their values of n_i, C_i and ΔV_i, as follows:

- Small n_i, Large C_i, large ΔV_i: For instance control signals, address buses and word lines that span a significant fraction of the width or height of the memory. Although relatively few in number, these signals have voltage swings equal to the supply voltage, V_{dd}.
- Large n_i, Large C_i, small ΔV_i: For instance, the bitlines in the memory array. Although the voltage swing on any single bitline is small, the number of bitlines more than makes up for this, so that the bitline power is the single largest component of the power dissipation in a typical SRAM.
- Very large n_i, Small C_i, large ΔV_i: This is the characteristic description of a logic circuit – very large numbers of logic gates, each with a small load capacitance, but a large voltage swing. An

SRAM contains relatively little active logic, and so this term is small compared to the power used for driving the long active signals in the memory.

Ways to Reduce Active Power

Based on this model, it is then possible to apply the appropriate power saving technique to each part of the SRAM, for instance by:

- Reducing the number of active long, high swing signals. E.g. by using wide predecode buses[5] and by routing combinations of control signals rather than multiple separate control signals.[6] This can mean that there are more long signal wires, but fewer of them are active in any given memory access, so that overall active power is reduced.
- Controlling the bitline voltage swing even in the presence of high process variability. As mentioned above, bitline active energy can be the single largest component of SRAM active energy. This is especially the case in situations of high process variability, where ΔV (the bitline voltage swing) can also be highly variable. SureCore has patented several techniques for managing bitline voltage swing in these situations in order to save bitline energy. One of these techniques (the Cascode Precharge Sense Amplifier – CPSA) also helps to reduce the static leakage current in the SRAM array.
- Prioritising the inputs to blocks of logic to minimise switching activity.

Leakage Power

Leakage currents – the tiny currents that still flow through inactive CMOS logic gates and memory bit cells – also need to be considered in a truly low-power SRAM. Although the leakage current in an individual bit cell is very small (in the picoamp or 10s of picoAmps range), when there are 10s or 100s of millions of bits on a chip the total current can become significant when either:

- There is a very large amount of memory on chip – 100 million bits x 10pA = 1mA.
- Long battery life is required – achieving multi-year life from a 1000mAh battery means average current must be in the 10s of microAmps range – comparable to the leakage from a few million bits.

As mentioned above, sureCore's patented Cascode Precharge Sense Amplifier controls the voltages on the bitlines in a way that reduces both active and leakage current in the SRAM array. It can typically reduce SRAM leakage current by 20% compared to memories that do not use the technique. In addition, our EverOn™ family of wide operating voltage memories (described below) have a more flexible set of low-leakage sleep modes than other memories. Instead of global active, light sleep, deep sleep, and shutdown modes that apply to a whole memory instance at once, EverOn™ memories have separate active, sleep, and shutdown states for each of 4 independent banks within the memory array, and also active and sleep states for the periphery circuits. This allows an application developer to fine-

[5] Within an SRAM, addresses are usually routed in a 1-hot format rather than a binary format. For instance, a 12-bit binary address can be converted to 6 'predecode' buses, each with a 1-of-4 format. Alternatively 4 predecode buses with a 1-of-8 format can be used – the latter uses more wires but has fewer active signals, and therefore uses less power.

[6] E.g. there is a choice between distributing a "start of cycle" and a "read/not write" signal, or distributing "start of read cycle" and "start of write cycle" signals. In the latter case, only one of the two can go high in a cycle, so that total active energy is reduced.

tune the sleep states of an application in order to minimise leakage. Figure 3 shows an example, where one quarter of an SRAM instance is active, together with the shared peripheral interface logic, while another quarter is asleep (in a reduced leakage state, but still retaining data) and the remaining half is shut down (with even lower leakage, but not retaining data).

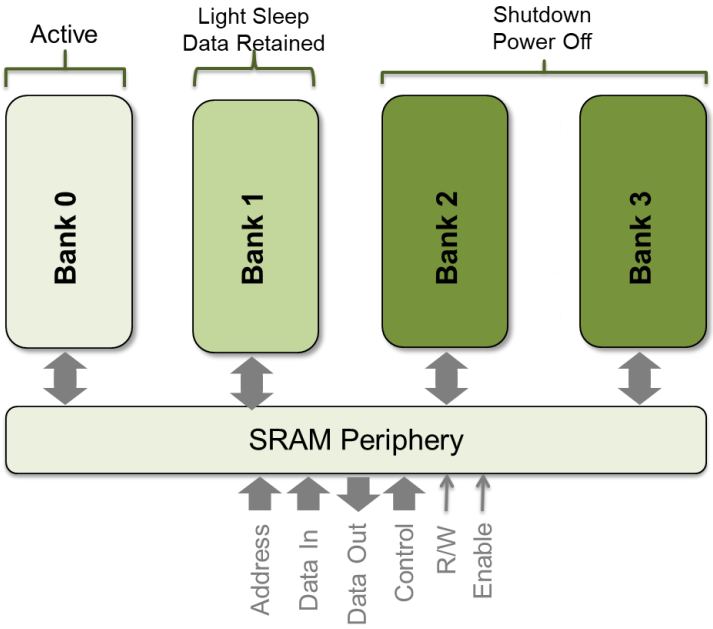

Figure 3 Independent banks in an EverOn™ SRAM

To achieve the same effect with traditional SRAMs with global active/asleep/shutdown modes would require using 4 separate small SRAM instances. This requires replicating the periphery circuits, which is an area penalty and also creates extra leakage paths in these replicated circuits.

SureCore's Low-Power SRAMs

The techniques described above have been used by SureCore to develop two families of low-power SRAM compilers: PowerMiser™ and EverOn™.

PowerMiser™

PowerMiser™ is a low-power SRAM compiler using the power management techniques described above. In both bulk CMOS and FDSOI process technologies it has demonstrated power savings of around 50% compared to other SRAMs running at the same voltage and frequency.

EverOn™

EverOn™ is a wide voltage range compiler, that adds a low V_{min} to the power saving techniques used in PowerMiser™. In many processes it has proved to be possible to have V_{min} approximately equal to V_{ret}. i.e. an EverOn™ SRAM does not just retain data at V_{ret}, but can also be accessed. EverOn™ includes very flexible read and write assist circuits in order to achieve this low voltage operation. In many applications that do not push DVFS to very low frequencies an EverOn™ memory can use the same supply voltage as its surrounding logic, allowing for simplified SoC power routing, and removing the need for level shifters between logic and SRAM running at different voltages.