Green Computing

Power-efficient memory and cache

Jothini Sritharan

Sommerakademie in Leysin, August 2016

Abstract

Initially, memory organization and the characteristics of caches is depicted. Reciting the main components, which are predominantly responsible for power-consumption, different methods for achieving a reduction in power-consumption are considered.

Power-efficient memories and caches can be constructed through an appropriate architecture by reducing accesses and dividing the construct into sub-segments, different encoding methods applied both on the bus and the words in a cache line and energy-efficient arrangement of a memories components especially transistors and capacities.

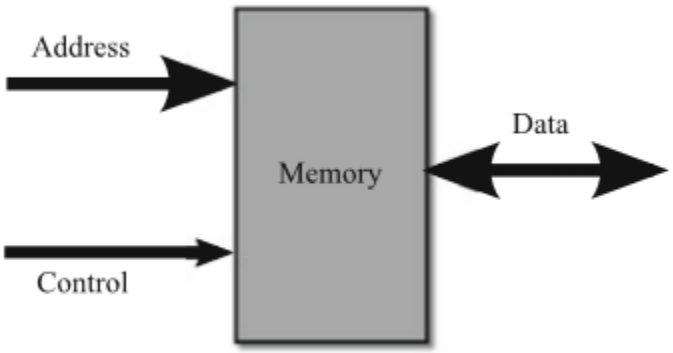
Contents

1 Introduction

In every type of electronic device or system the storage of data plays an important role. In form of memory and caches information can be saved in different ways. However, transmitting, reading, writing or erasing data consumes energy and therefore it is necessary to find methods which reduce the power consumption of memories and caches.

2 Memory organization and caches

2.1 Memory organization

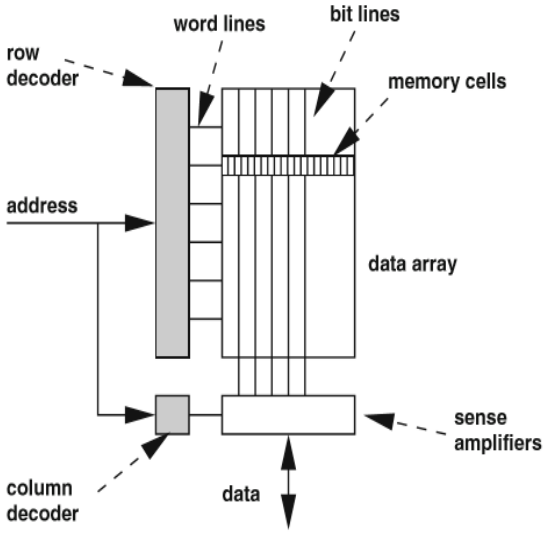


The interaction with the memory takes place in connection with a bus-system. The bus-system can be divided into input and output.

The address-bus, which belongs to the input, points at the location of the memory which is accessed. The number of memory locations destines the size of the address-bus.

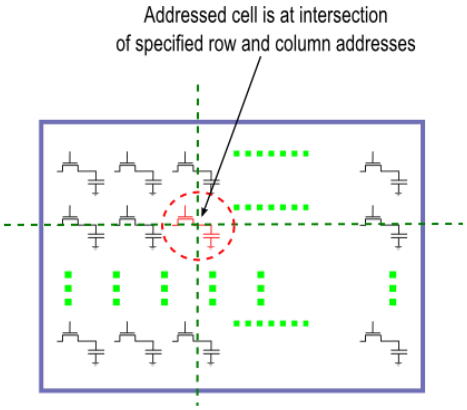
The control – another input – serves as an indicator for important data. Furthermore, it marks whether it is a reading or a writing operation.

The data-bus carries required data and it can be used as both input for a writing operation and output for a reading operation. When using the reading operation, the data-bus delivers as output the referring location with the required data, whereas the writing operation repurposes the data port into an input and stores data at the chosen memory location.



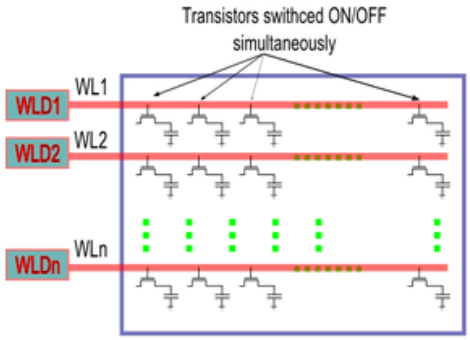
The memory itself can be considered as two-dimensional matrix. Based on this matrix, the reading operation can be structured into three steps. Firstly, the row address, which contains the higher order bits, is decoded and as a result activates the corresponding word line. In step two, the word line selects a row of cells, which causes a transfer of data between the cells and the bit lines, which can be imagined as a pipe carrying information. Finally, the sense amplifier receive the data from the bit lines. In the last step, the column decoder uses the least significant part of the address, the column address, to choose the bits in the right column from the selected row and transfers the addressed bits to the data bus.

2.1.1 Memory-organization using the example of Flash memory



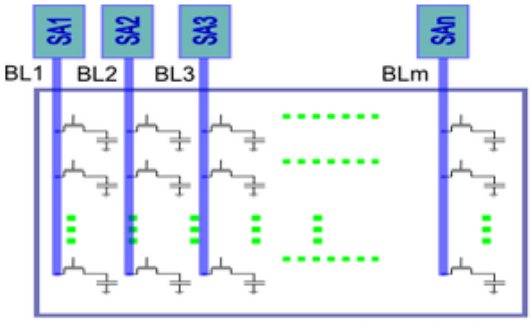
Flash memory belongs to the non-volatile memory, which means that if the voltage is turned off, the data still remains.

A two-dimensional array of memory cells made of transistors makes up a memory-unit on a flash memory chip. An array is a data structure, which comprises similar elements. In order to access one specific element of the array, you have to address the two-dimensional array with a row- and a column-number. The rows are known as word-lines as mentioned above and the columns constitute the bit-lines. The array consists of many transistors, which are placed in the columns and rows. Every transistor represents a bit as it can occupy two states: conductive and insulating.



The word-lines connect the gates of all transistor in a row. When now a specific word-line is activated that means when the voltage on it is turned on, then all the transistors in this row will open simultaneously.

The word-line-drivers are responsible for applying a voltage in a row. If there are many transistors in a row, which implies that the row's capacitance also increases, the word-line-driver needs to be more powerful in order to reach every transistor in this row. This stands in contrast to the array efficiency as a more powerful word-line-driver needs more space on the chip and thus reducing the space for the array.



The columns in an array, the bit-lines, connect the drain/sources of each transistor in a column and therefore carrying the information. Information is read from the cell or written into the cell by the sense-amplifiers - each bit-line is connected to a sense-amplifier. Analogous to the word line drivers, the size of sense amplifiers and thus its performance is physically limited by the size of the chip.

A compromise has to be made.

2.2 Caches

Contradictory requirements referring to the memory system, which demand on the hand a large storage for a vast amount of data and at the same time the access delay and energy dissipation should be held on a low level, are known as "memory wall". The memory hierarchy poses a solution to this problem: In general, it can be said that large memory and higher memory levels lead to a corresponding delay to run through all the data and also more energy per access. A processor consists of the areas register, cache memory, main memory, and secondary memory. Once a data is accessed for the first time, it can be put into a lower memory level, which results in a shorter delay and lower energy dissipation when it is accessed for a further time. The cache memory is in charge of storing recently accessed data and instructions. In opposite to the register file, the cache can consist of several levels, where the lower levels are usually located within the chip next to the processor and higher levels could be located outside the chip. This leads to the

fact that data is frequently found on the lowest level. A finding of data is called “Cache hit”, whereas an unsuccessful search is known as “Caches Missing” which initiates to search in the next level for the required data. It can be distinguished three types of Caches Misses:

Compulsory misses happen, when data is accessed for the first time which implies that it never had the chance to be stored in the cache. Capacity misses arise from the limited size of a cache. When the working data is larger than the cache itself the cache data may be replaced. Finally, conflict misses happen as a consequence of the access time constraints by the system, which lead to limitations of the ability of trouble-free searching and replacing memory.

The cache missing ratio is a benchmark to indicate to what extent the accesses could not be serviced from the cache and have to continue with the next memory level. It can be calculated as number of Cache misses divided by number of Cache accesses.

When a cache miss happens, a bunch of memory words are transferred between cache and main memory, which is known as cache line. The question is how does the mapping between the memory address and cache location take place? First of all, the main memory is structured into blocks of the cache line size. Different types of functions can be discerned:

Direct-mapped cache	Two-way set associative cache	Fully-associative cache
The function of direct-mapped cache matches cache locations to a memory block address according to the equation. Cache Line = (Block Address) modulo (Cache Size)	It reduces the cache conflicts of the direct-mapped cache by adding constituents in each line of a cache set. Nevertheless, the searches need to be improved and take place more often to ensure if a data element exists in the cache → higher energy dissipation	Any memory-block can be matched to any cache location, as long as its sum is smaller than the cache. No conflict misses occur, but capacity misses may take place. → but longer access time

Additional regulations help structuring administration in a cache. For example, the Least Recently Used (LRU) policy replaces the cache line first which has not been used for the longest time.

3 Energy consumption concerning memory access

The following three main components basically determine the energy consumption during memory access as it requires a lot capacity within the wires:

1. address decoders and word lines

2. data array, sense amplifiers and the bit lines

3. the data and address buses leading to the memory

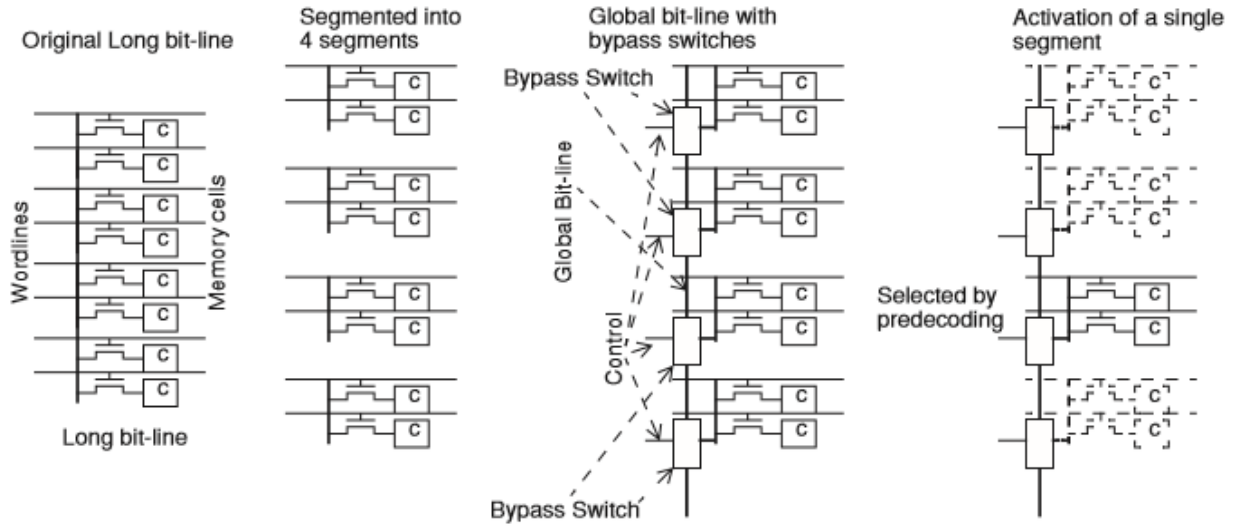
To mitigate energy dissipation, these three categories have to be issued to achieve a reduction of power consumption.

4 Ideas of reducing power consumption

4.1 Power-efficient memory Architectures

4.1.1 Partitioned memories and caches

The memory array is divided into several banks, which leads to shorter bit lines and therefore smaller transferring paths and shorter delays.



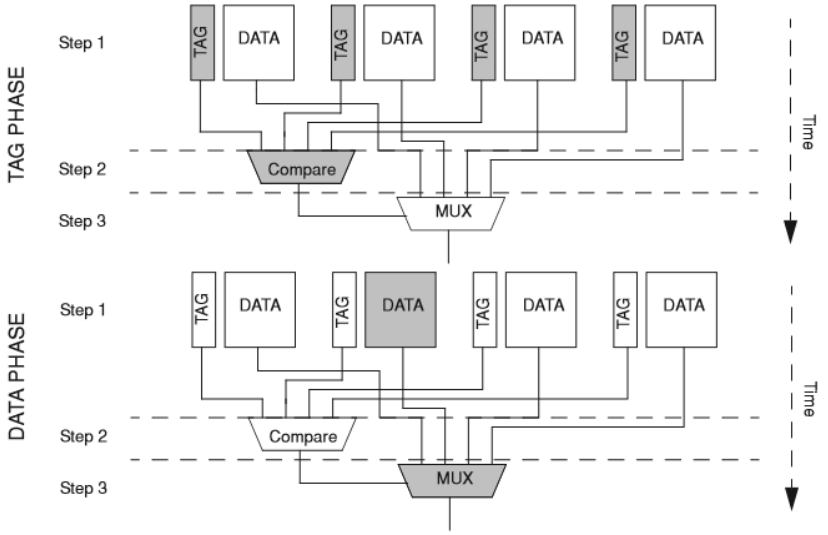
Large caches are divided into multiple subarrays. Thus the bit-lines and the word-lines are splitted into smaller segments avoiding excessive delays of long wires and leading to shorter transferring ways. A global bit-line connects the segments and each part can be activated by switches. Redundant ways can be disabled, which means that they are not activated and therefore not passed by. The capacitive load of the whole global-line is now less than that of the original long bit-line. Smaller sense-amplifiers and drivers are sufficient.

4.1.2 Additional memories

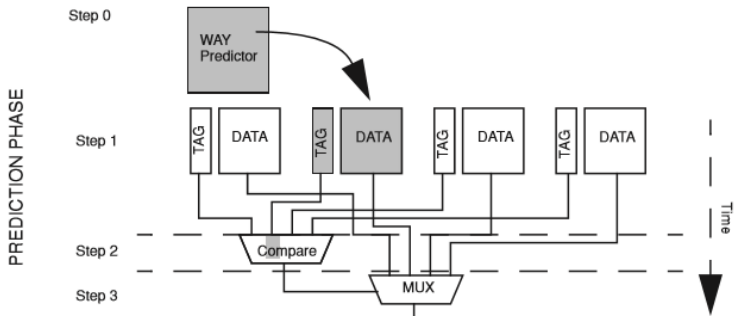
A small extra cache, which stores recently used data and serves as a buffer, is searched first and may prevent browsing through the whole main cache. The shorter this buffer is and the more a cache hit happens, the more successful this concept is.

4.1.3 Reducing tag and data array fetches

The aim is to reduce the number of accesses by only searching the tags without data first because in a conventional cache both tag and data is browsed through at the same time. The process is divided into two steps: Finding the matching tag and then picking only the data of the right tag. There is also the possibility to predict a potential tag, which may reduce the accesses enormously if a cache hit occurs. This is also known as phased cache.

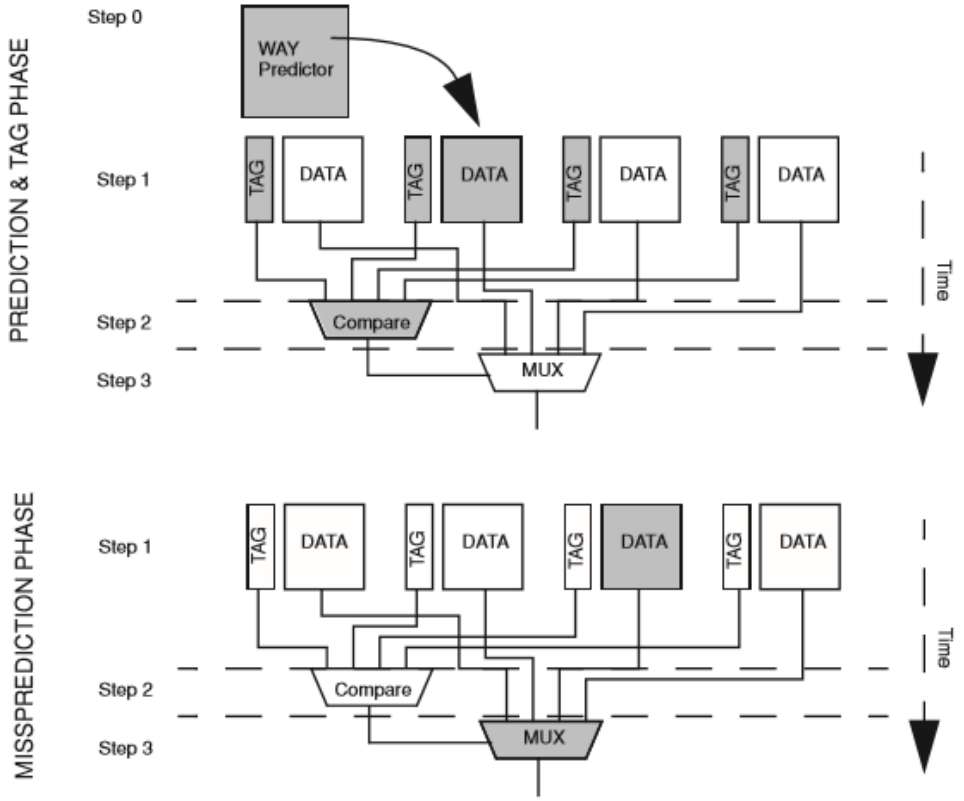


This method can be expanded by an algorithm, which determines the Most Recently Used data, which has a high probability to be used again and therefore is examined first. If a cache hit occurs, everything is fine and if a miss occurs, then all other tags are compared according to the phased cache method.



These two methods can also take place at the same time: In the first step all tags are examined and only the data of the tag, which is chosen by the predictor is also looked at. If this was a wrong

prediction, then the Data of the right tag, which was filtered during the tag Phase in step one, is transferred to the next step.



The disadvantage of the way-predicting methods is that when a cache miss happens that means a wrong prediction is made then a second round is needed, which costs additional time and power.

4.1.4 Reducing cache leakage power

Another possibility is to turn off the power of the lines, which have not been accessed for a specific time.

4.2 Translation Look-aside Buffer (TLB)

The Translation Look-aside Buffer is a cache, which is used in connection with virtual memory. Virtual memory was developed to create a continuous, linear storage area to relieve a programmer from managing physical memory. It leads to a unification of different memory sources. The CPU generates virtual addresses, which are divided into pages and later translated

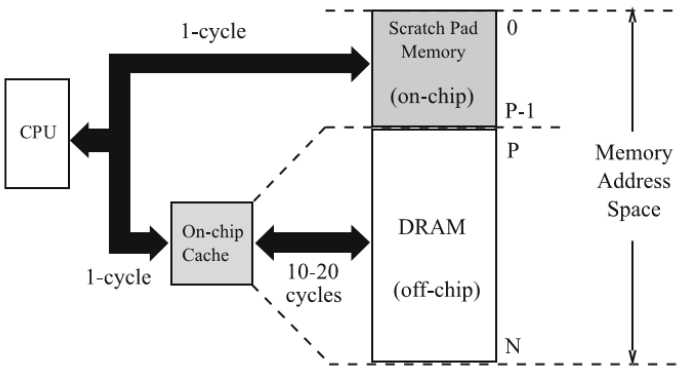
into physical address by using a translation table. The Translation Look-aside Buffer now stores the translation information from virtual to physical address of the most recently accessed memory locations.

4.3 Memory customization

To improve performance and power-efficiency the hardware architecture can be adapted to the respective case of application. Different parameters like cache size, line size etc. can be varied within the cache memory.

4.4 Scratch pad memory

Scratch Pad memory is a kind of memory which is managed by the compiler or programmer. They help reduce system power by avoiding tag searches associated with caches. Due to the fact that data remains on the SPM until it is manually removed the access time is now more predictable. In the first cycle the SPM and the Caches are browsed through and when a cache miss occurs the DRAM is searched.

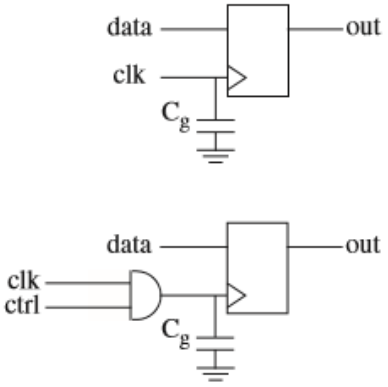


More frequent and smaller data are predominantly stored in the SPM.

Dynamic management ensures that an unneeded array in the SPM is replaced by another required array.

4.5 Clock-Gating

Clock-Gating means in general to turn off the clock-pulse in a part of a circuit in case it is not needed to save power. In every clock-cycle the capacitance Cg of the circuit is charged and discharged. When the circuit remains in the same state it is a waste when the capacitance is charged. To solve this problem the clock is connected with a control signal by an AND-gatter and if the clock is temporarily not needed the control signal



is set on 0 which results in the AND element not conducting. This method is used for example in notebooks.

4.6 Idle-width switching activity: Core

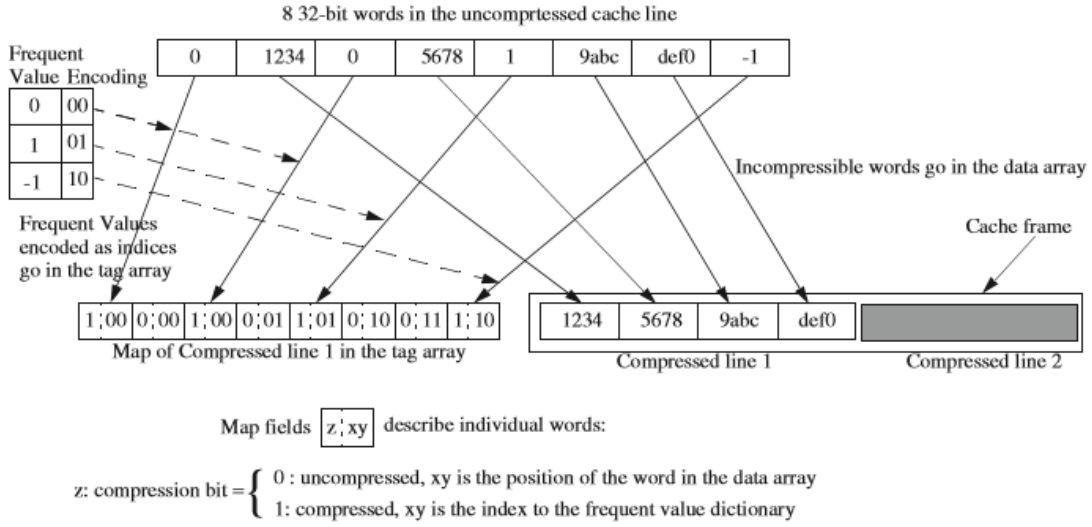
We often encounter 32- or 64-bit processors. But most of the used operations only require 8 or 16 bit, which means that the majority of the provided bits are unused and only power waste. To counteract this issue there are two different methods: Firstly, put many small operations together so that they almost fill the 64 bit. Or disabling the unused bits and eliminating the part of the hardware which does not carry significant bits. These two methods can also be combined and coexist.

4.7 Cacheable switching activity

Another method of reducing power is to avoid repetitive computing activity. That means for example when executing a program and calculations are conducted for a specific instruction, these calculations have to be done again when the same instruction appears again. So the result of the computing can be stored in a cache. Is the instruction needed again then the cache is browsed through and repeated calculation avoided which saves power. Results of Activities which can be outsourced are for instance loop iterations in a program. Of course the loop needs the same input in order to produce the same result which then can be stored in a cache.

4.8 Idle-width switching activity: Cache

Different packing methods of data enables a compression of the information and thus offers more space for other data on the freed space. In addition, reading now only fewer bits leads to power savings.



In the initial state the uncompressed cache line consists of eight 32-bit words, which occupies a lot of storage. The cache line places both compressible and uncompressible words. The compressible words are the frequent values like 0,1 and -1. They get a new index like a translation, which represents them. The uncompressible words are put unaltered in the data array. This data array also keeps the information in form of a Map, in which sequence the words are positioned originally and which word is compressed and which not. A map-field consists of two elements. The first one can be a 1, which means the data is compressed, like the first word of the cache line and it can also be a 0, which means the data is not compressed, like the 1234 which is on second place in the cache line. The second element of a map field either delivers the index of the compressed word or the position of the uncompressed word. For example, the first word, the 0, is compressed and the 00 represents the index of this word. Whereas the fourth word, the 5678, which is uncompressed, has a 01, which means that it is located at second position from the uncompressed words.

The method seems to complicate the cache line but it actually leads to power savings as the whole information with position is stored in this array and eight 32 bits are no longer needed.

4.9 Bus Encodings

A data bus energy consumption depends on two factors: Firstly, the wires capacitance and then the number of signal transitions and transfers on the wire. So the carried data influences the energy consumption on a bus. That is why it is worth it to deal with encoding methods.

One target area are the busses that carry the addresses. Since an address is made up of two components, the high-order component, which in most cases only transmits redundant information, and the low-order component that changes frequently. In this case the high-order component can be stored in a cache, whereas the low-order part is always transmitted from processor to memory.

Referring to the encodings, there is on method called bus-inversion: Data is transferred either in its original form or when the distance is way smaller then in its compliment form. Of course, an additional signal needs to be sent additionally in order to mark that this data is not the original one but needs to be reinverted at the end.

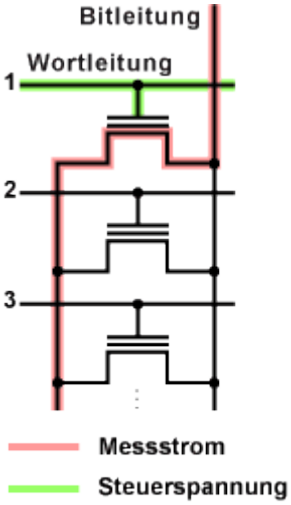
5 Low Power Memory Design

One decisive way of reducing power is lowering the operating voltage in memories. At the same time the memory operation must not be interfered in its functionality. In the following different memories are seen from this perspective.

5.1 Flash-Memories

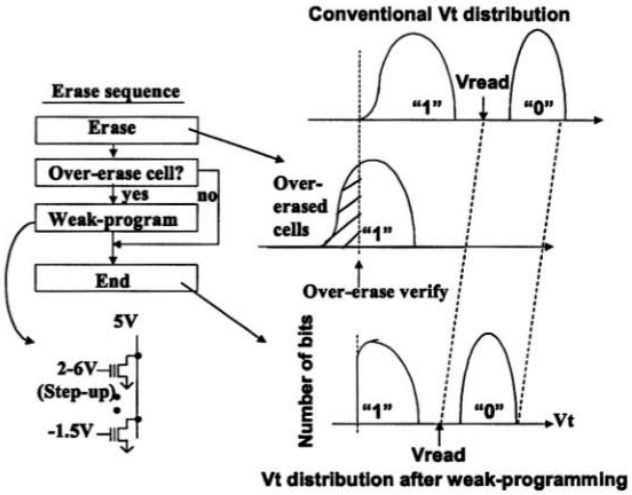
Due to the fact that flash memories can keep the data even if the extrinsic voltage is shut down the stand-by power is almost zero and explains the importance of flash-memories in regard to energy-efficiency. There are two groups of Flash-Memories: The first one aims at a fast random access that means easy memory access like the NOR-Flash-Memory and the other one at high-bit-density like the NAND-Memory, which is used for high-capacity data storage.

5.1.1 NOR-Flash-Memory

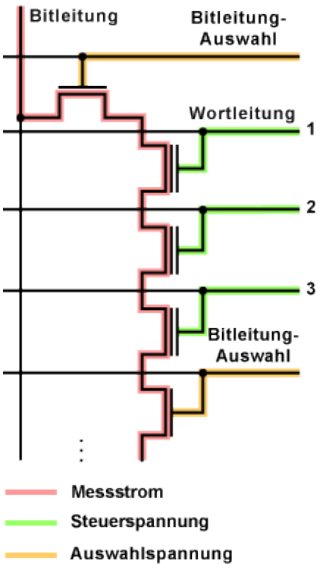


Due to the fact that the cells are parallel connected each cell can be addressed separately.

To lower the reading voltage, the bit-by-bit weak program uses a potential, which is lower than the ordinary voltage, which finally leads to an offset of the whole curve. On the x-Axis you can see the Voltage and now the reading voltage is lower than without the bit-by-bit weak program, submitting a reduction in voltage and thus power-consumption for generating this low voltage.

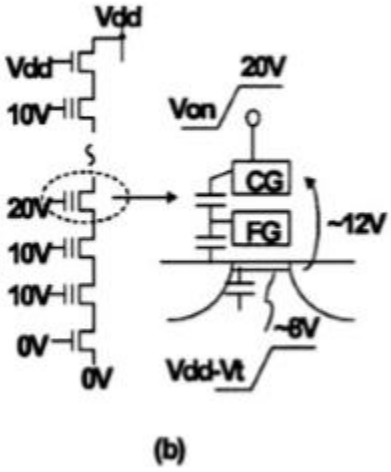


5.1.2 NAND-Flash-Memory



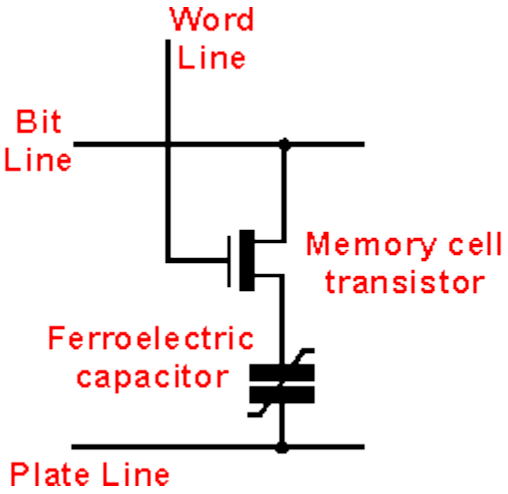
The series-connection of the cells requires reading and writing in blocks. Because of the few data lines, the NAND-Flash occupies less space than the NOR.

Self-boosted programming technique

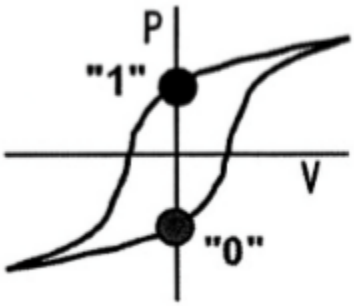


The voltage of the bit-line voltage is the supply voltage. The unselected transistor is precharged which leads to a rise in the selected word-line voltage so that the self-boosted channel voltage also rises because the select transistor is cut off.

5.2 Ferroelectric Random Access Memory (FeRAM)

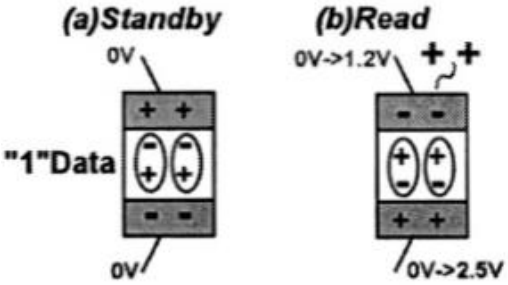


The transistor has the purpose to choose the required cell, whereas the ferroelectric capacitor changes it polarity depending on a voltage impulse and therefore leading to a storable state. During the reading-process another voltage impulse is sent over the cell and depending on a change or constancy of the polarization direction a different current is measurable, which indicates in which kind of state the cell has been before. So information is stored in form of polarization in a ferroelectric layer.

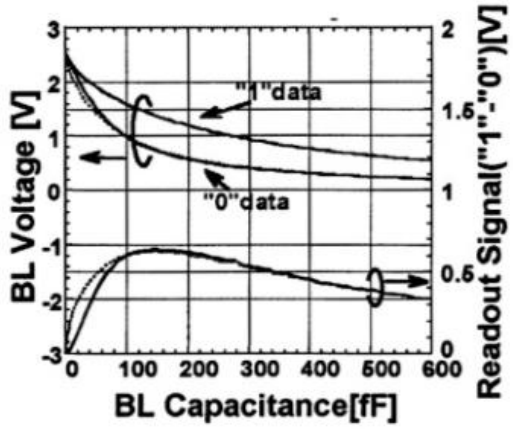


These are the two stable states the cell can occupy by having a different polarization at the applied voltage of 0.

The picture illustrates that during the reading process the data is changed and needs to be restored.

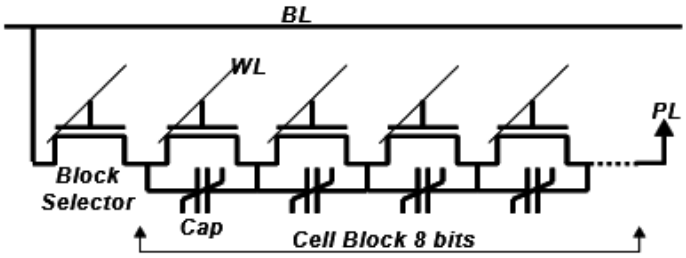


5.2.1 Low-Voltage FeRAM design



In order that the reading operation functions appropriately a clear distance between the state of 0 and 1 is needed so that the two states are explicitly distinguished. The diagram shows that the Bit-Line Voltage and thus the readout-signal, which is the difference between the 1 and 0 signal, depends on the used capacitance. By choosing the adequate capacitance the memory can be optimized.

5.2.2 Chain FeRAM

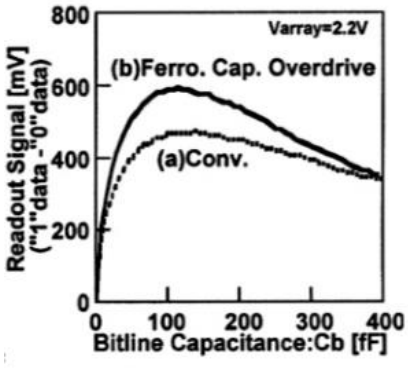


In contrast to an ordinary FeRAM, where the constant capacitors are connected in parallel and therefore adding their capacitances, the Chain FeRAM connects the cells in series, making a chain block with the selecting transistor. Due to the fact that the plate line is now shared by several cells, the available space is used efficiently and the plate line driver can be enlarged because of the now obtainable space to deliver enough power.

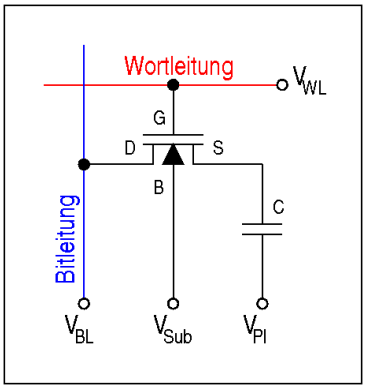
5.2.3 Other low voltage techniques

Because a FeRAM's functionality depends on the level of polarization it is necessary to aplly a high enough voltage to achieve a strong enough polarization. On the other hand, high voltage also means more power-consumption. To solve this problem one method divides the bit lines into several sub bit lines, which are still connected to the main bit line with separation transistors. This causes that only the sub bit lines, which are needed are used and the odd sub bit lines remain unused and therefore avoid wasting charges.

Another method, the so called **ferrocapacitor overdrive**, pulls down an additional capacitance in the series connection of capacitances and enables a rise in the readout signal, which equals better polarization.



5.3 Embedded DRAM



The embedded DRAM consists of a selecting transistor and a capacitance, which saves the data. When a high enough voltage is applied on the word line, the connection between source and drain becomes conductive and the one end of the capacitance is connected with the bit line, which can now put charge on the capacitance. Due to the simple architecture the embedded DRAM can achieve a high data band width on a small space and reduce Input/Output power consumption. Precharging helps supporting low-voltage eDRAMS and thus saving power.

6 Summary

To sum it up, it can be said that caches and memories contribute in a decisive way to power-consumption in electronic devices. Thus it is necessary to make them more power-efficient. By optimizing a caches architecture, adding extra memories with most recently used data and adapting the components, especially transistors and capacitances, in a memory the power-consumption of caches and memories can be considerably reduced.

Bibliography

Power-Aware Design Methodologies by Pedram, Massoud, Rabaey, Jan M

Computer Architecture Techniques for Power-Efficiency by Stefanos Kaxiras and Margaret Martonosi

Power-Efficient System Design by Panda, P.R., Silpa, B.V.N., Shrivastava, A., Gummidipudi, K.

http://www.elektronik-kompendium.de/sites/com/0610041.htm (07.08.2016)

https://de.wikipedia.org/wiki/Dynamic_Random_Access_Memory (06.08.2016)

https://en.wikipedia.org/wiki/Flash_memory (06.08.2016)