Quantitative Optimization and Early Cost Estimation of Low-Power Hierarchical-Architecture SRAMs Based on Accurate Cost Models

Yuan Ren, Tobias Noll

Quantitative Optimization and Early Cost Estimation of Low-Power Hierarchical-Architecture SRAMs Based on Accurate Cost Models

Yuan Ren and Tobias Noll

Chair of Electrical Engineering and Computer Systems
RWTH Aachen University, Aachen, Germany

{ren, tgn}@eecs.rwth-aachen.de

Abstract. Dedicated low-power SRAMs are frequently used in various system-on-chip designs and their power consumption plays an increasingly crucial role in the overall power budget. However, the broad amount of choices regarding the capacity, wordlengths and operational modes make it hard for designers to determine the optimal SRAM architecture. Additionally, many low-power techniques and circuits are frequently utilized but not supported by previously proposed cost models. In order to solve these problems, a cost-model based quantitative optimization approach is proposed. In particular, a fast and accurate power estimation model is built for aiding the low-power SRAM designs. It precisely fits the various complex SRAM circuits and architectures. The quantitative approach provides useful conclusions early in the design phase guiding further optimizations. The estimation error of the power model has been proven to be less than 10% compared to results based on time-hungry extracted-netlist simulations in a 40-nm CMOS technology.

Keywords: SRAM, Power Model, Quantitative Parameter Optimization

1 Introduction

SRAMs are widely used in many applications as caches etc. due to their fast access speed but also contribute significantly to area cost and power consumption. Particularly in system-on-chip design, dedicated SRAMs with optimized architecture and circuits are often applied for achieving low-power. The optimization of those dedicated SRAMs for lowest possible power at a given performance is a quite challenging task because of the complexity of the design space. An attractive approach is to perform a quantitative optimization based on cost models. Such cost models not only support the optimization process but also allow for early cost estimation in the system

conception phase. The focus of the study is laid on the power cost considering the increasingly significant role of power consumption of SRAMs.

From the low-power perspective, SRAMs with hierarchical architecture, as described e.g. in [1][2][3][4], are very attractive choices. A quantitative optimization approach for the hierarchical-architecture SRAMs deserves further research.

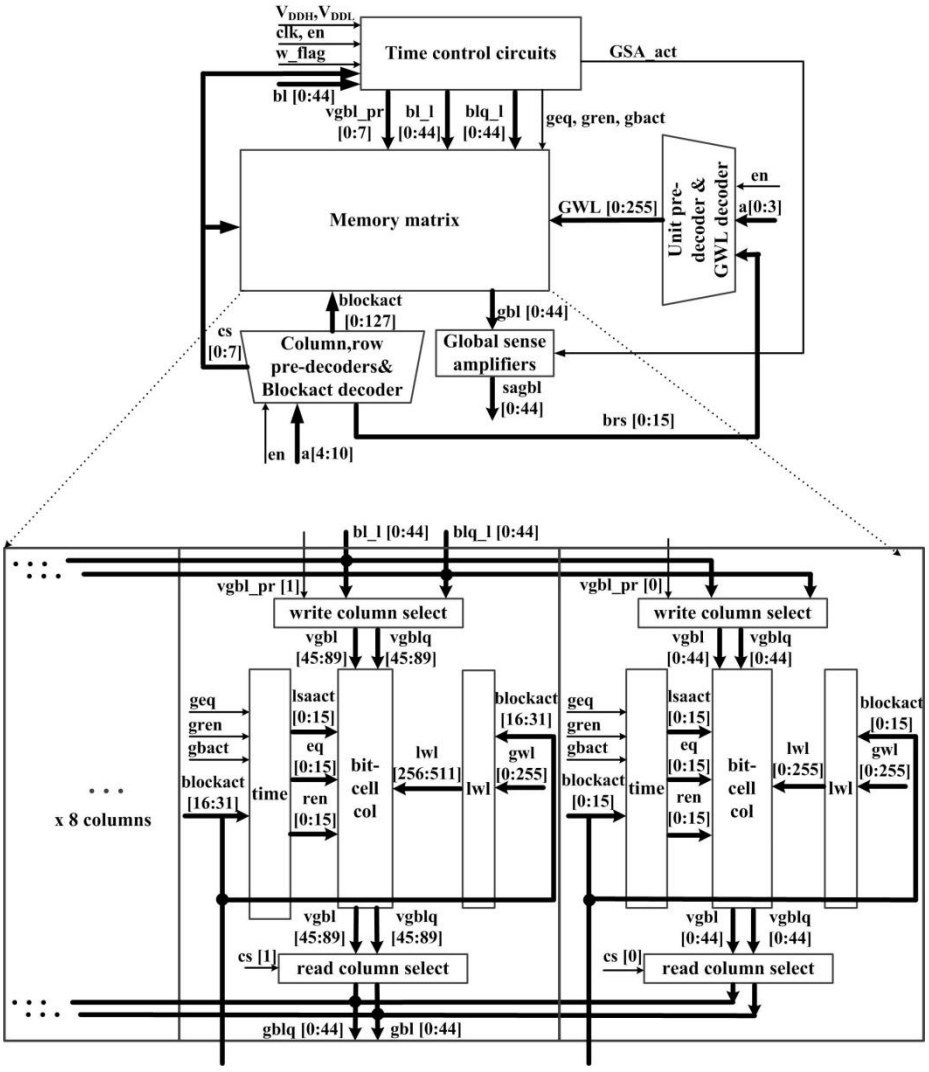

Fig. 1. A block diagram of a conventional on-chip SRAM with a hierarchical architecture

Table 1. Estimation approaches investigation

	[6]	[7]	[8]	[9]	[10]	This Work
Hierachical using LSA	Y	N	N	Y	N	Y
Specific efficient Circuits	N	N	N	N	N	Y
No Reference Design Needed	Y	Y	Y	N	Y	Y
Validation Technology	≥32nm	130nm	65nm	N.A	45nm-180nm	40nm

Fig. 1 provides a block diagram of such a hierarchical-architecture SRAM of 2K words with a 45 bit-wordlength. Apart from the timing control circuits, the memory matrix and the address decoders dominate the total power consumption. These two components exhibit a large design space regarding the underlying architecture and circuits. In the hierarchical architecture the memory matrix is typically organized in 2^m ($m=3$) columns. Each column includes a local timing generator, a bit cell column and a local wordline decoder. A bit-cell column consists of 2^n ($n=4$) local blocks and a local block is sub-divided into 2^u ($u=4$) words vertically. Thereby, the long bitlines and wordlines are both divided into global and local lines for reducing the switched capacitances. Moreover, the use of local sense amplifiers further reduces the power consumption by decreasing the signal swing on the long interconnects. Furthermore, the bit cell column could consist of various efficient circuits, such as assist circuits [1], stable bit cell and bit-interleaved technique [2] and pre-charge schemes [3][5]. Apparently a large design space exists for selecting hardware-efficient architectures and the underlying circuits. Especially for SRAMs with different capacities and features, the time to market constraints make design space difficult to be explored. Therefore, there is a strong demand to come up with a cost model, by which architecture parameters, local circuits and power reduction techniques are characterized and quantitatively analyzed.

Many available power cost models were investigated for hierarchical-architecture SRAMs using various low-power circuits and techniques. As it is illustrated in Table 1, the widely used CACTI tool [6] is a tool to understand large caches in the context of microprocessors. It focuses on microprocessor caches (including cache coherency techniques) and with fewer possibilities for choices of circuits and techniques, in [1][2][3][5]. For on-chip SRAMs CACTI is found very inaccurate due to its incompatibility with specific circuits and techniques so that it cannot be used as an optimization or design guiding tool. The power model in [7]cannot be used for SRAMs with low-power architectures containing divided bitline and divided wordline structures. Moreover this, it cannot deal with a variety of efficient specific circuits. In [8] only the traditional subdivided bitline structure is discussed without considering other possible low-power architectures or circuits. Moreover they do not consider LSAs in their model since the LSAs significantly contribute to the leakage power. The approach in [9] requires a complex reference design of a whole SRAM whose characterization is time-consuming, which makes it not practical for designers at the early design stage.

Moreover, neither LSA in a hierarchical architecture nor energy-efficient circuits for local blocks are included. The power model in [10] discussed a binary tree SRAM based on the approach in [7] which makes it similarly inappropriate. Moreover, the energy consumed by long interconnects existing in the binary tree organization is not considered in the total energy consumption in a proper way. Hence, these models cannot help designers in making quantitative decisions about architecture and circuits.

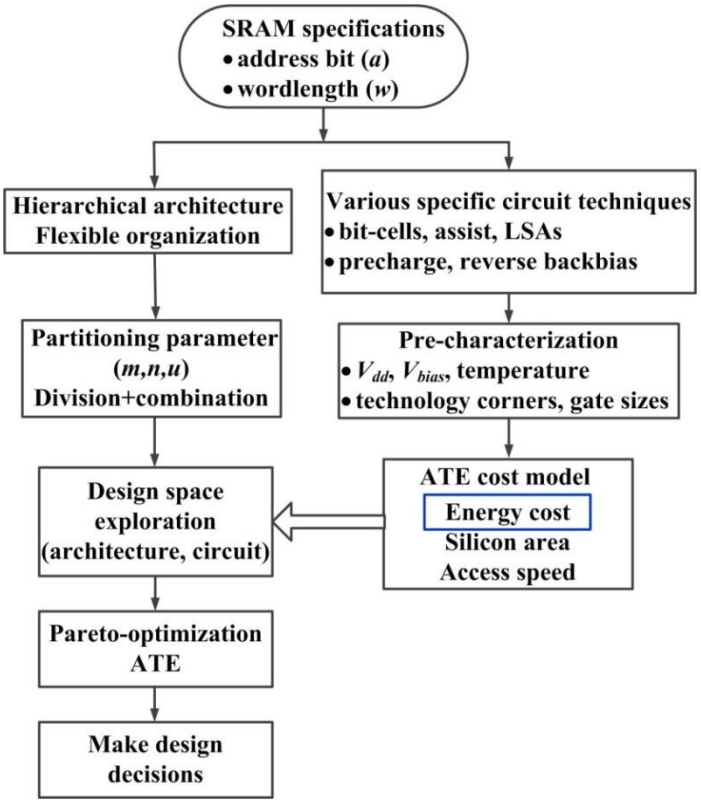

Fig. 2. Overall flowchart of the cost estimation environment

A predictive and mature design flow for on-chip SRAMs must simultaneously consider energy (E), area (A) and speed (T) properties. For this reason a cost estimation environment is under elaboration serving as an efficient design aid and a quantitative optimization tool. Fig. 2 sketches the overall flowchart of this environment. The SRAM specifications are given by the capacity in number of words and wordlength. The first task is to determine the optimal architecture and most effective circuit techniques. A hierarchical architecture is taken as the subject to be analyzed and optimized. The architectures are specified by the partitioning parameters (m, n, u). Here, parameter m is the column address decoder width, which denotes the number of columns ($M=2^m$) and n is the row address decoder width, which denotes the number of rows ($N=2^n$) in a memory matrix. Parameter u is the unit address decoder

width, which denotes the number of cells ($U=2^u$) in a column unit. On the whole, the three parameters define the partitioning and organization of a hierarchical-architecture SRAM with a total capacity of $M*N*U$ words being equal to 2^a address bits with wordlength (w). The decomposition and combination possibilities of the three parameters are explored for selecting the optimal architecture partitioning.

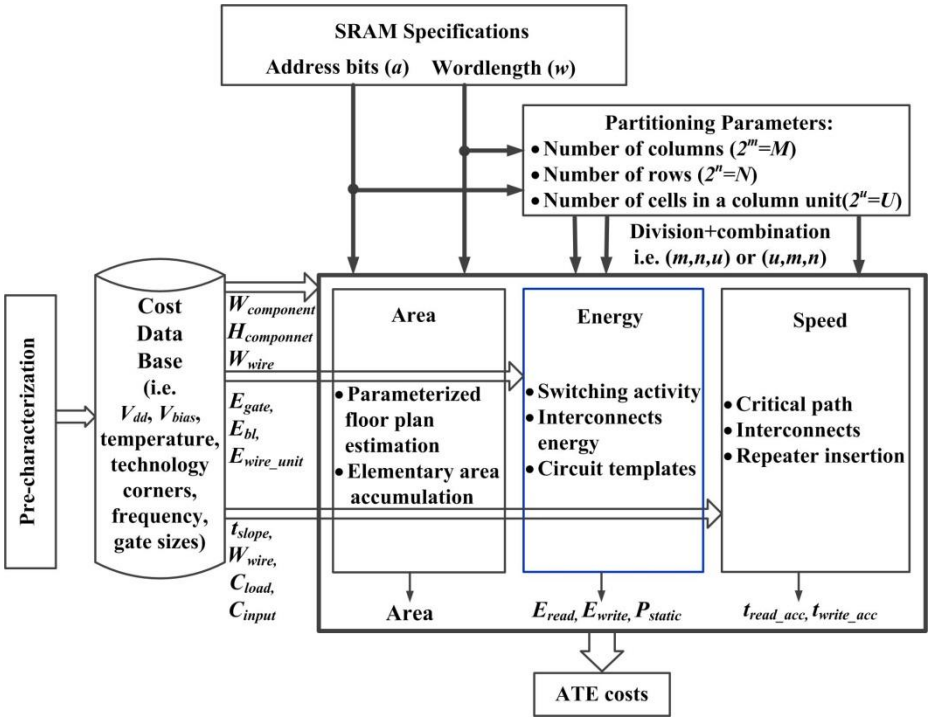

Fig. 3. Overview of an ATE cost model

Besides the choices related to partitioning regarding the design space, there exists a wide choice of various energy-efficient circuit techniques. These circuit techniques, such as the ones related to the bit cells, assist circuits and local sense amplifiers (LSA) cause additional overhead and complexity to the overall design. The consumed energy is difficult to evaluate and estimate since it depends on the context in which these circuits are used. Their non-standard features cannot longer be characterized using the available models which only include standard features. Hence there must be a quantitative benchmarking tool to assess these circuits and techniques. Accordingly, their respective power consumption should be estimated and compared while using different configurations so that the optimal one can be selected. A pre-characterization methodology is employed for capturing their distinct features and quantifying the analysis. With these 2 sets of inputs, partitioning parameters and specific circuit techniques, a design-space exploration is carried out, while building a cost (A, T, E) model for on-chip SRAMs. This way a Pareto-optimization is carried

out by trading off the three costs (A, T, E). Finally the design decisions regarding architecture and circuits are made.

The idea of a cost-model based pre-characterization of elementary components (i.e. logic gates and bit cells) is further explained in Fig. 3. Given specification parameters (a, w), all the possibilities of the partitioning parameter (m, n, u) are generated and then these possibilities are evaluated by the presented cost model. An ATE cost database is built by proper circuit simulation of extracted component netlists, which depends on the use case (e.g. V_{dd}, V_{bias}), technology corners, temperature, frequency and gate sizes.

- The area can be estimated by a parameterized floor plan estimation and an accumulation of elementary widths and height values (e.g. $W_{components}$, $H_{components}$).
- In the energy model, the elementary energy values of basic circuits (e.g. E_{gate}) are accumulated with the partitioning parameters and switching activity probabilities. The interconnect energy is estimated from the wire length and the energy per unit length (E_{wire_unit}). Finally the total energy consumption is derived by an accumulation of elementary energy (e.g. E_{bl}) from all used circuit components.
    - The speed can also be derived by using the cost data base (e.g. t_{slope}, W_{wire}, C_{load}, C_{input}) and a elementary delay accumulation along the critical path involving long resistance-capacitance interconnects.

    In this contribution, we focus on the energy cost model and the relevant optimization approach. In the proposed model, only a few necessary basic circuit components (i.e. basic bit cells) need to be characterized. These circuit components could be verified by a number of Monte-Carlo simulations for ensuring robustness. Moreover, the pre-characterization including simulation time and model building only requires couple of hours and afterwards the estimation results can be acquired in a few minutes. Therefore, the total effort of the cost model is much less compared to a complete reference design.

2 Hierarchical architecture

A conventional hierarchical low-power SRAM organization including address decoders and memory matrices is illustrated in Fig. 4. It is partitioned into $2^{(m+n)}$ blocks which are organized as an array of $M=2^m$ block columns and $N=2^n$ block rows, while m and n are dependent on the decompositions of column and row address decoders. The outputs of these two pre-decoders, 2^m column select (CS) and 2^n row select (RS) signals, are further decoded by NOR or NAND gates for generating the $2^{(m+n)}$ Blockact signals for selecting a specific block. Each block is composed of $U=2^u$ words placed vertically and it includes w column units. The column unit is considered as a basic unit for the memory matrix, which includes 2^u cells per column unit and one LSA. A unit decoder and the row decoder generate $2^{(u+n)}$ global wordlines (GWL) at the output to access the selected row in the block. Afterwards, the GWLs and Blockact signals are

further decoded to $2^{(m+n+u)}$ local wordlines (LWL) to access the selected word. In this way, bitline and wordline capacitances are reduced for meeting the low-power requirement. Also, the introduction of LSA reduces the voltage swing on global bitlines.

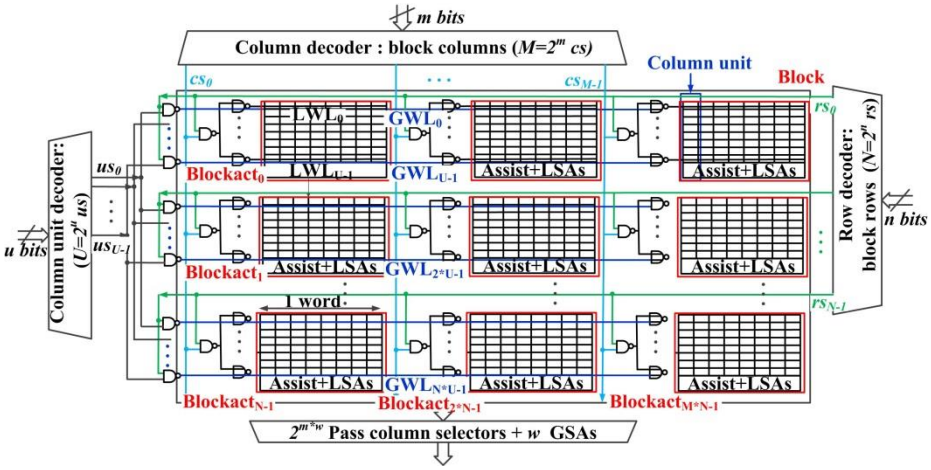

Fig. 4. Hierarchical-architecture SRAM organization comprising DWL structure

As discussed before, the memory matrix is organized into $M=2^m$ block columns. In these block columns various complex but efficient circuits are utilized. In Fig. 5, a block column is shown for exemplifying these specific circuits. It is composed of a local timing-control signal generator, a LWL decoder and a bit cell column. In the local timing-control signal generator, the global timing pulse signals are combined with the Blockact signals to generate local timing pulse signals for the $N=2^n$ blocks. Similarly in the LWL decoder, the GWL signals running through the whole memory matrix are combined with the Blockact signals to generate LWL signals for each word in the bit cell column. The bit cell column is instantiated by high-threshold voltage bit cells with reverse back-biasing long channels, equalizer pre-charge scheme, read/write assist circuits and a wide local sense amplifier [5]. A block in a bit cell column is composed of w column units. Each column unit includes $U=2^u$ 6T cells, assist circuits and a LSA.

The choice of m and w defines the parasitic wordline capacitances and the wordline structure. Either a non-divided wordline (non-DWL) or a DWL structure can be selected according to the capacity and wordlength. The parameter n defines the bitline hierarchy and therewith affects the global and local bitline capacitances. Especially, charging and discharging the bitlines contributes significantly to the overall power consumption. The number of cells u in one column unit determines to a large extent the minimum energy consumption for one operation. The n and u must be carefully selected for trading the least frequent use of LSAs and the minimum switching capacitances.

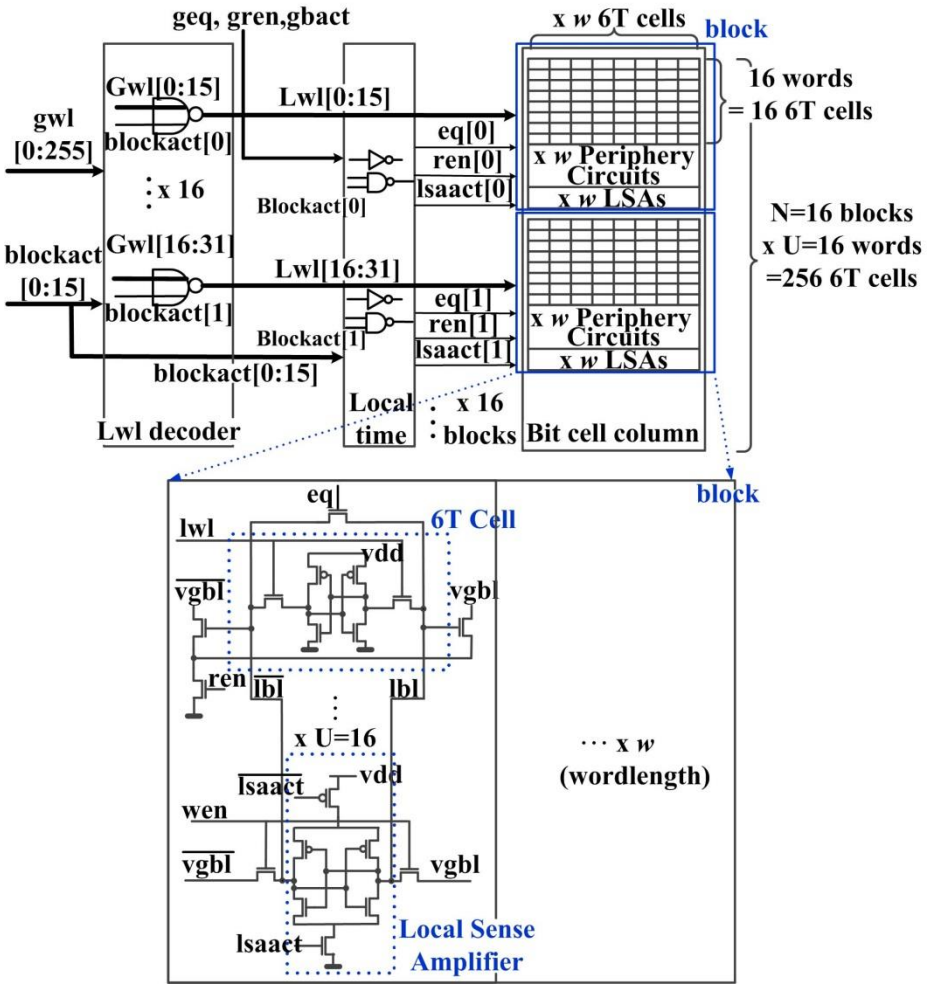

Fig. 5 Specific circuits found in a column of a hierarchical-architecture SRAM macro

3 Partitioning impact analysis

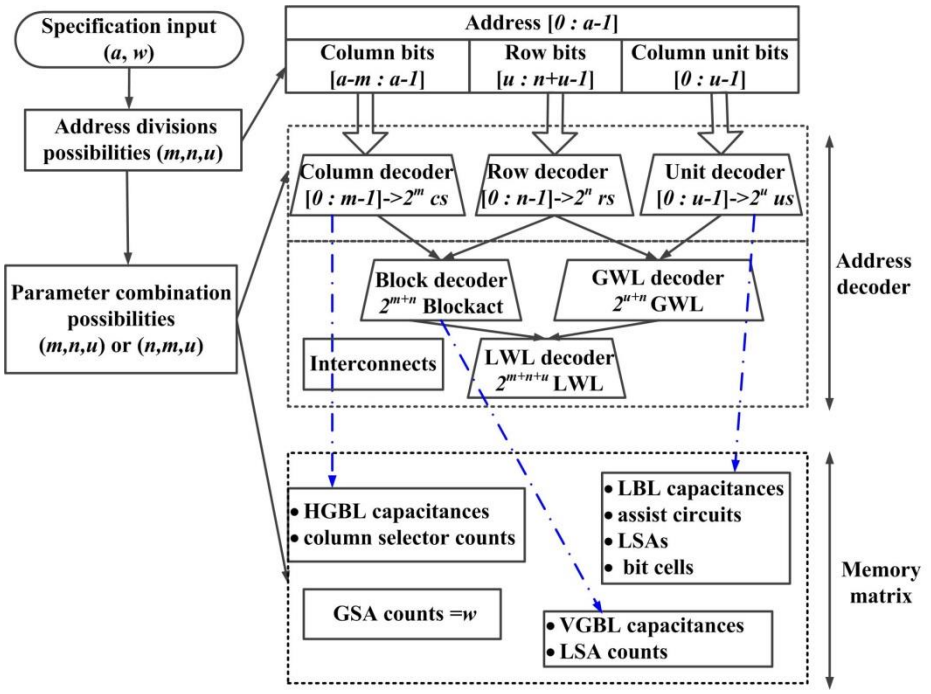

Fig. 6. Partitioning parameter possibilities and their impact on SRAM components

SRAMs typically include two major contributors to power consumption: the address decoders and the memory matrices. In the hierarchical architecture (Fig. 4), the way of dividing and combining the address decoders determines how the memory matrix is partitioned into sub-blocks. A probabilistic estimation approach is employed for estimating the switching activity and power consumption of the address decoder especially regarding whether or not a distributed wordline structure is used. The memory matrices including complex assist and periphery circuits, which consume a large portion of power, were also modeled and characterized. Four basic circuit templates and a power estimation method are proposed to extract and describe the architecture and circuit characteristics for the hierarchical architecture. The specific circuits used within the four circuit templates can be altered without changing the estimation approach itself. Various power reduction techniques, e.g. precharge schemes in [3][5], circuit techniques in [1][2][5], can be pre-characterized and benchmarked in the same configurations, which makes this model very appropriate for customized SRAM designs.

For an SRAM with a address bits comprising a capacity of 2^a words with a word-length w, a flow chart of the power estimation model is shown in Fig. 6. The two portions constituting the power model are the address decoder and the memory matrix. For the address decoder, a address bits are divided into three sections (m, n, u), which are decoded by the three pre-decoders including column, row and unit decoder.

In the 2nd stage, a block decoder combines RS and CS signals to generate the Blockact signals. A word-row decoder uses RS and US signals for generating the GWL signals. In the 3rd step, a word decoder uses both Blockact and GWL signals to produce the LWL signals. The three decoders are all composed of NAND or NOR gates and are arranged in a matrix-like select circuit. The sum of the power consumed by the pre-decoders and the matrix-like select circuits provides an estimate of the total power estimation of the whole address decoder. For the memory matrix, the parameters (m, n, u) are used for quantifying and analyzing the bitline and wordline structure. Since the parameters represent the number of the sub-modules and determine the final SRAM architecture, their respective impact is analyzed and attributed to the components of the memory matrix. The parameter m determines the capacitances of the horizontal global wordlines (HGBL) and the amount of pass transistors used as column selectors. The number of the Blockact and GWL signals affects the capacitances of vertical global bitlines (VGBL) and GWLs respectively. Finally, the choice of u has an impact on the power consumption from LBLs of accessed cells, assist circuits, and LSAs. Therefore, a quantitative analysis about the dependency relations is made between the combinations of (m, n, u) and all the power contributors. Given the specifications a and w, all the possible partitioning parameters (m, n, u) are evaluated by estimating the respective power consumption. Finally, the optimal parameter selection is saved for fulfilling different ATE design requirements, such as minimum power consumption.

4 Power model of address decoder

4.1 Basic circuits of address decoder

As shown in Fig. 6, the address decoder includes three pre-decoders and three distributed decoders. The three pre-decoders can be decoders comprising either a large fan-in or a small fan-in, depending on their input numbers. The other three intermediate decoders are regarded to be matrix-like select circuits which are composed of logic gates distributed in a matrix. A probabilistic method is employed for modeling the underlying switching activities of these logic gates, by which the transition power consumption of the matrix-like select circuit is estimated. The large fan-in decoder is composed of a matrix-like select circuit and two small fan-in decoders. Therefore, if the energies associated with small fan-in decoders and basic gates are available, the energy of the three pre-decoders and the three distributed decoders can be derived by the probabilistic method. Also, a realistic topology estimation approach is used to estimate the wire capacitances and area of different (m, n, u).

A circuit pre-characterization database is built in the pre-characterization phase, which includes the related configuration regarding the use case (V_{DDH}, V_{DDL}), process corners, temperature and frequency. The database can be acquired in a short time since the complexity of the basic circuits is much less compared to the overall SRAMs. Moreover, such a pre-characterization approach is also convenient for estimating the static power dissipation. Small fan-in decoders are usually very flexible

Table 2. Characterization database of basic decoders (TT, 25 °, 400MHz, V_{DDH}=0.9V)

	Decoder 2-to-4	Decoder 3-to-8	Decoder 4-to-16
Dynamic Energy(aJ)	2105	4400	8460
Static Power (nW)	12	31	56
Input Capacitance(fF)	1.44	2.44	3.93
Width(µm) Height(µm)	2.73 2.02	3.45 3.53	4.17 5.63

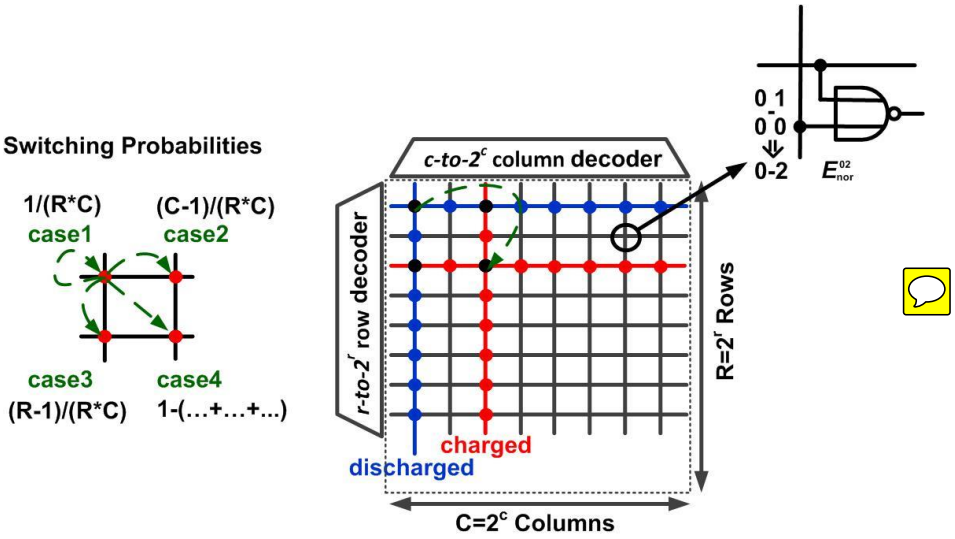

Fig. 7. Four switching cases and their switching probabilities in a distributed decoder

and customized regarding their layouts and transistors, so these basic circuits were simulated based on extracted-netlists. Dynamic energy, static power, input capacitances and areas are listed in Table 2 for TT corner, 25°C, 400MHz and 0.9V supply in a 40-nm CMOS technology. Dynamic energy figures were obtained from random-input power simulations. Other corners were evaluated as well, but only TT corner numbers are reported in this chapter. The static power values were determined by power simulations at different frequencies and approximately linear extrapolation of the results for *P(f=0)*.

4.2 Switching activity

Besides the energy of the basic decoders (Table 2), the matrix-like select circuits formed by NAND or NOR gates also contribute significantly to the total power consumption. Such circuits are typically acting as distributed decoders and are located

among the memory matrices. As illustrated in Fig. 7, a distributed decoder composed of NOR gates has an aspect ratio given by *R* rows and *C* columns. When another address is accessed, not only the corresponding gates switch but also the other gates in the row and column are charged and then discharged. As different transitions of each gate lead to different amounts of consumed energy, the corresponding energy of each NOR gate in its transition cases must be estimated separately. Additionally, for the overall matrix-like circuits composed of NOR gates four switching cases (Fig. 7) exist. For each switching case the energy and switching probability are also derived.

Table 3. Energy of NOR gate for all possible input transition possibilities (TT, 25 °, 400MHz, V_{DDH}=0.9V, V_{DDL}=0.3V)

Inputs transitions	**00**	**11**	**12**	**13**	**01**	**02**	**03**	**23**
Total Energy(aJ)	11	4	400	68	213	320	235	68
Inputs transitions	**22**	**33**	**21**	**31**	**10**	**20**	**30**	**32**
Total Energy(aJ)	2	3	315	228	705	655	880	228

In Table 3, a subset of the database for the consumed energy of the basic NOR gates is shown for each possible input transition. E.g. the transition 00→01 of a two-input NOR gate is depicted by the decimal equivalents 0→1. Hence, a transition 10→01 is depicted by 21 and its switching energy is denoted by E_{NOR}^{21}, and so on. In particular, the static power of the four "no transition" situations (00, 11, 22, 33) is also included in Table 3, where the total energy is obtained when the frequency is set to 400MHz. A NAND matrix-like select circuit can be pre-characterized and estimated in the same way as well.

For a distributed decoder with (*R* x *C*) NOR gates four possible switching cases exist. *Case1* means no switching of the selected column or row. *Case2* means a switching of the selected column within the same row. *Case3* means a switching within the same column. *Case4* means a switching from one gate to another gate located in a different row and a different column. In order to elaborate the switching details and its energy distribution, *Case2* is exemplified in four steps as shown in Fig. 7. Since the switching happens in the same row, it means that one cross point in the matrix is selected and another one in the same row is unselected. a) Hence, the selected NOR gate switches from 1 to 0 and another unselected one switches from 0 to 1. b) Horizontally in the selected row, (*C-2*) NOR gates have no switches and their inputs stay at 1. c) Vertically, (*R-1*) NOR gates switch from 2 to 3, which means they are discharged in the relevant column. Also, (*R-1*) NOR gates switch from 3 to 2 which means they are charged in another column. d) The remaining NOR gates do not switch and stay at 3. Using the transition energy depicted in Table 3 the respective energy of the four switching cases is derived as

$$E_{DecMatrix}^{Case1} = E_{NOR}^{00} + (R-1)\cdot E_{NOR}^{22} + (C-1)\cdot E_{NOR}^{11} + (R-1)\cdot(C-1)\cdot E_{NOR}^{33} \quad (1)$$

$$E_{DecMatrix}^{Case2} = E_{NOR}^{01} + E_{NOR}^{10} + (R-1)\cdot\left(E_{NOR}^{23} + E_{NOR}^{32}\right) + (C-2)\cdot E_{NOR}^{11} + (R-1)\cdot(C-2)\cdot E_{NOR}^{33} \quad (2)$$

$$E_{DecMatrix}^{Case3} = E_{NOR}^{02} + E_{NOR}^{20} + (C-1)\cdot\left(E_{NOR}^{13} + E_{NOR}^{31}\right) + (R-2)\cdot E_{NOR}^{22} + (R-2)\cdot(C-1)\cdot E_{NOR}^{33} \quad (3)$$

$$E_{DecMatrix}^{Case4} = E_{NOR}^{03} + E_{NOR}^{12} + E_{NOR}^{21} + (R-2)\cdot\left(E_{NOR}^{23} + E_{NOR}^{32}\right) +$$

$$E_{NOR}^{30} + (C-2)\cdot\left(E_{NOR}^{13} + E_{NOR}^{31}\right) + (R-2)\cdot(C-2)\cdot E_{NOR}^{33} \quad (4)$$

In particular, the four cases occur with different probabilities, which are associated with the number of rows and columns. These probabilities may also depend on the way the memory is used in an application but in the context of this chapter the focus is set on the random accesses. Assuming a random address access pattern for the SRAM, the probabilities are derived as follows. Dynamic energy of the matrix-circuit is estimated as

$$E_{matrix}(R,C) = E_{DecMatrix}^{Case1}\cdot\left(1/(R\cdot C)\right) + E_{DecMatrix}^{Case2}\cdot(C-1)/(R\cdot C) + E_{DecMatrix}^{Case3}\cdot(R-1)/(R\cdot C) +$$

$$E_{DecMatrix}^{Case4}\cdot\left(1-1/(R\cdot C)-(R-1)/(R\cdot C)-(C-1)/(R\cdot C)\right) \quad (5)$$

The equation was verified using several different combinations of rows and columns and shows 5% estimation error compared to extracted-netlist simulation results.

4.3 Energy cost related to interconnects

As technology keeps shrinking the role of interconnects becomes increasingly significant in the total power budget. Particularly interconnects incur large capacitive loads in the dense SRAM layout. As described in Fig. 1, the 1st stage pre-decoders and the 2nd stage decoders are typically placed around the memory matrix. The 3rd stage LWL decoders are distributed into the block columns. Hence, the aspect ratio of the LWL, local timing circuits and the bit-cell column must be considered together. For estimating the associated interconnect lengths, a floor plan containing the dominating memory matrix and address decoder must be determined in advance.

Since different layout floor plans result in different wire and coupling capacitances, two typical placements are selected as possible layout organizations. In the reference floor plans, a matrix circuit is always much larger than the other two sub-blocks. Therefore, a compact topology exhibiting smaller area is selected. As shown in Fig. 8, a horizontal placement leads to different interconnect lengths compared to a vertical placement. For a large fan-in decoder, two pre-decoders and a matrix-like select circuit are placed in both ways for evaluation purposes. For the given floor plans the total area, interconnect lengths and wire capacitances are estimated and compared. By assessing which placement is more compact for the overall floor plan, the two arrangements are selected.

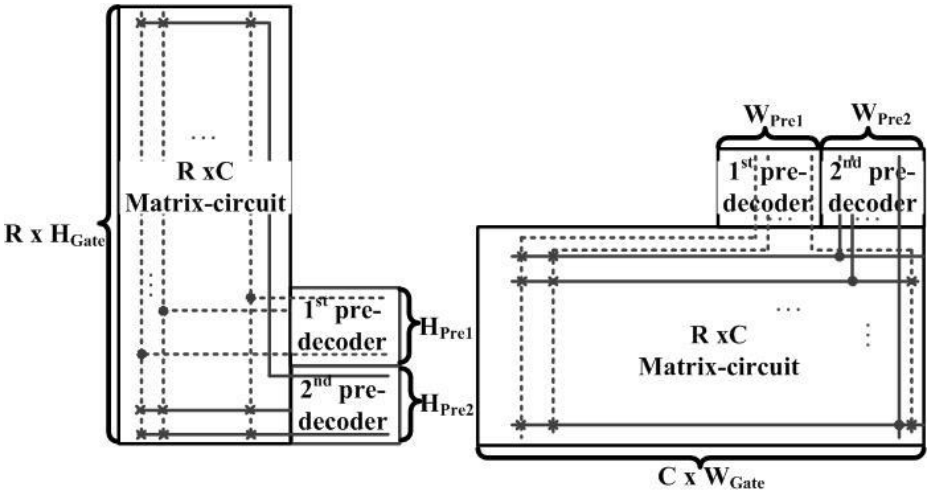

Fig. 8. Two empirical placement orientations for wire capacitance estimation

The height and width of the two pre-decoders are denoted as (H_{pre1}, W_{pre1}) and (H_{pre1}, W_{pre2}), which are obtained from the pre-characterization given in Table 2. The height and width of basic gates (H_{Gate}, W_{Gate}) such as NOR or NAND gate are also available. Given the two placement possibilities, the height and width of the required wiring can be derived for horizontal (H_h, W_h) and vertical (H_v, W_v) placements respectively.

$$H_h = H_{Gate} \cdot R,\ W_h = W_{Gate} \cdot C + W_{Pre1} + W_{Pre2} \tag{6}$$

$$H_v = H_{Gate} \cdot R + H_{Pre1} + H_{Pre2},\ W_v = W_{Gate} \cdot C \tag{7}$$

As the wire lengths in the two sub-blocks are much shorter compared to the matrix circuits, the following criterion is used to select the more compact topology. If

$$abs\left(H_{Pre1} + H_{Pre2} - H_{Gate} \cdot R\right) < abs\left(W_{Pre1} + W_{Pre2} - W_{Gate} \cdot C\right) \tag{8}$$

holds, the placement should be horizontal, otherwise a vertical placement is applied. Subsequently, the wire lengths can be estimated by computing the amount of gates and their individual sizes in the selected placement. The two floor plans can be used either for a large fan-in decoder or a memory matrix and its surrounding circuits. For a global SRAM floor plan (Fig. 1), the two pre-decoders are replaced by a LWL decoder and a local control timing generator. The matrix-like select circuits are replaced by a memory matrix column. Thereby, a floor plan of a block column is determined. The decision procedure to estimate the global wire capacitances is similar.

Considering the switching activities of the relevant wires the energies for switching the interconnects in the two possible topology scenarios are estimated as

$$E_h(R,C) = Vdd^2 \cdot 0.08 \cdot [W_h \cdot (C-1) + H_h \cdot (R-1)]/(R \cdot C)$$

$$+ Vdd^2 \cdot 0.08 \cdot (W_h + H_h) \cdot (R \cdot C - R - C - 2)/(R \cdot C) \quad (9)$$

$$E_v(R,C) = Vdd^2 \cdot 0.08 \cdot [W_v \cdot (C-1) + H_v \cdot (R-1)]/(R \cdot C)$$

$$+ Vdd^2 \cdot 0.08 \cdot (W_v + H_v) \cdot (R \cdot C - R - C - 2)/(R \cdot C) \quad (10)$$

The wire capacitance per unit length of a metal wire is assumed as an appropriate value (*0.08fF/μm*) for a 40-nm technology. This value is evaluated and modified when coupling capacitances exist in very dense layouts. Moreover, under the assumption that only the column decoder switches and the row decoder does not, the switched capacitances are only determined by the width (W_h) with a switching probability *(C-1)/(R C)*. In case that only the row decoder switches and the column decoder does not, the switched capacitances considering its switching probability are equal to *0.08* H_h *(C-1)/(R C)*. If both row and column decoders are switching, the switched capacitances are computed considering both width and height. To summarize, for the two typical floor plans wire lengths and capacitances are estimated, which leads to a decision regarding which floor plan has to be assumed.

4.4 Verification of address decoder estimation model

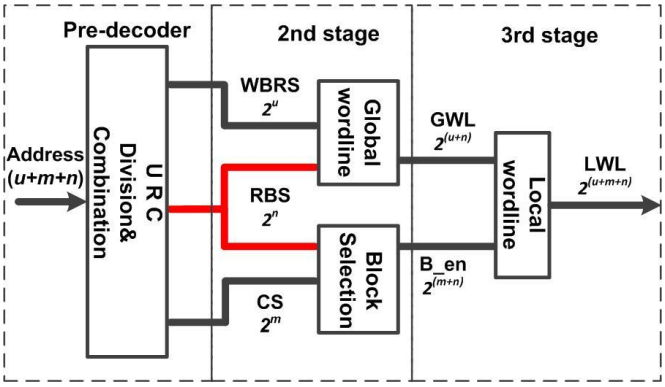

Fig. 9. Divided wordline (DWL) structure for address decoder

For low-power SRAMs with large capacities and long word length, it is inevitable that a DWL structure is superior to non-DWL. Because in the non-DWL structure long wordlines suffer from the half-select problem and numerous bitlines are needlessly precharged. As shown in Fig. 9, in a DWL structure the block row decoder which generates 2n RS signals is used twice instead of only once in non-DWL structure. Also an extra distributed GWL decoder is used in a DWL structure which brings additional area cost. But the benefit of a DWL structure is that switching occurs within a smaller memory matrix and thereby the total energy is significantly reduced.

Therefore this is a tradeoff between the power and area of the address decoder and memory matrix.

For the energy estimation of the DWL-address decoder, large fan-in decoders with *(n+u)* inputs can be handled by a nested loop calculation using smaller fan-in decoder data from Table 2 and the relevant matrix-like selected circuits. The energies of distributed decoders in the 2nd and 3rd stage are estimated by the approach described above. The dynamic energy and static power figures are derived as

$$E_{dyc\_tot} = E_{dyc}(n) + E_{dyc}(u) + E_{matrix}(N,U) + E_{wire}(N,U) + E_{dyc}(m) + E_{matrix}(M,N) + E_{wire}(M,N) + E_{matrix}(N \cdot U, M) + E_{wire}(N \cdot U, M) \quad (11)$$

$$P_{sta\_tot} = P_{sta}(n) + P_{sta}(u) + P_{sta}(m) + P_{sta\_matrix}(N,U) + P_{static\_matrix}(M,N) + P_{sta\_matrix}(N \cdot U, M) \quad (12)$$

The dynamic energy of three pre-decoders are represented by $E_{dyc}(n)$, $E_{dyc}(u)$ and $E_{dyc}(m)$. The parameters *m, n* and *u* denote the input widths of the three decoders respectively. The energy can be acquired from Table 2 (optionally in combination with small fan-in decoders and matrix-like circuits). For the second stage, energy figures for word-row and block decoders are given by $E_{matrix}(N,U)+E_{wire}(N,U)$ and $E_{matrix}(M,N)+E_{wire}(M,N)$. $N=2^n$ and $U=2^u$ represent the number of rows and columns of the matrix-circuit. Note that in the 3nd stage a matrix ($N\,U$, M) is applied instead of a matrix ($N\,U$, $M\,N$) since every GWL signal only needs 2^m Blockact signals to select the word in that column, cf. in Fig. 9. For an address decoder in a non-DWL structure there is no use of GWLs. Therefore, the energies from $E_{matrix}(M, N)$ and $E_{wire}(M, N)$ are not counted into the total energy.

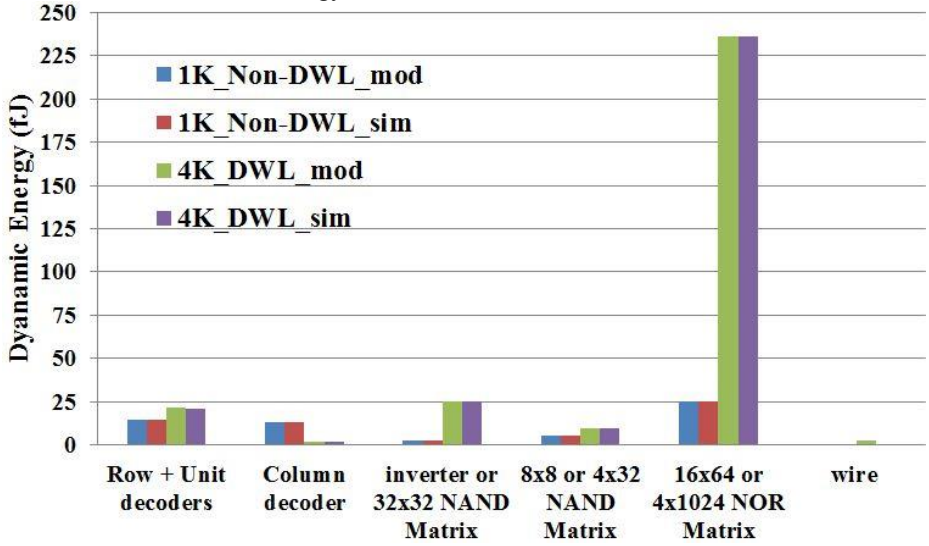

Fig. 10. Simulation v.s. estimation energy for a 1K Non-DWL and a 12-to-4K DWL decoders

Fig. 10 shows the simulated energy versus the estimated energy of a non-DWL-1K address decoder and a DWL-4K address decoder. The breakdown energies are compared accordingly. In particular, the energy associated with wiring capacitance is quite low compared to other components. This is explained by a short global interconnect length and non-significant coupling. For larger address decoders and denser layouts, the energy contribution from interconnects cannot be neglected any more. It can be seen that the 4x1024-NOR-Matrix circuit dominates the overall power for the 4K DWL decoder. The comparison indicates that the estimation errors of the address decoder power model are less than 10%.

5 Power model of the memory matrix

The contribution of the memory matrix to the total memory access energy is dominated by the cycle-based pre-charge and discharge of long bitlines. For low-power memory matrix designs, assist circuits, bit cells and pre-charge schemes span a large design space complicating the power modeling. Their complex features bring significant influences on the layout placement location and the switching capacitances. Accordingly, the total energy cannot be computed by directivity accumulating their respective individual energies. Additionally, the use of LSAs in [1] brings low-voltage swing at global bitlines and high-voltage swing at local bitlines. The complexity with multiple VDD plays at larger scale which makes it more difficult to estimate the power consumption. As before, the variable partitioning parameters (m, n, u and w) result in different access gates and parasitic capacitances due to different wire lengths. Another challenge is that read, write and standby operations must be considered separately, including a hierarchical bitline structure and the memory cell toggling state. In order to solve these issues, four circuit templates are proposed to act as a black box for pre-characterization. In this way a database depending on the use case (V_{DDH} and V_{DDL}), technology corners, temperature and the characteristics of gates (width) and wires is generated. Finally, the elementary energies from assist circuits, bit cells and vertical global bitlines are separated by our estimation approach. Combined with the partitioning parameters the power consumed by the overall memory matrix is estimated accurately. Leakage power is estimated in a similar way.

5.1 Four circuit templates

Four circuit templates based on the circuits given in [5] are presented as basic circuit elements for characterizing the complex assist circuits and specific bit cells. As shown in Fig. 11, a single cell circuit template is presented first to separate the elementary energy from multi cell circuits. Its dynamic energy consists of contributions from the local bitline of each cell (E_{lbl}), the local wordline (E_{lwl}) and the periphery circuits including precharge circuits (E_{pre}), read/write assist transistors (E_{ren}/E_{wen}) and LSA (E_{lsa}). For pre-characterization the customized design a layout for the single cell circuit template is drawn. This way the dynamic energy (E_1) and static power ($P_{static1}$) are obtained by extracted-netlist simulation

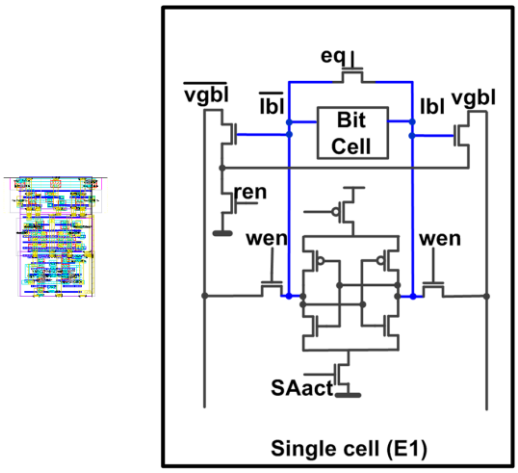

Fig. 11. A single cell circuit template

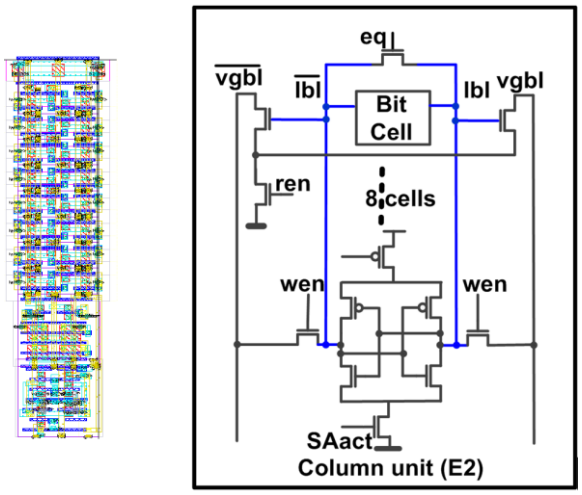

Fig. 12. A column unit circuit template

$$E_1 = E_{pre} + E_{ren} + E_{lwl} + E_{lsa} + E_{lbl} \,. \tag{13}$$

For separating the elementary energy from the local bitline of each cell (E_{lbl}), a column unit circuit template is drawn in Fig. 12. Its dynamic energy (E_2) and static power ($P_{static2}$) are also obtained from extracted-netlist simulation. In the same way its total energy (E_2) is decomposed to several elementary energies. By taking the LSA and periphery circuits apart, the energy from the 1-8 cells are linearly interpolated. Thereby, the energy consumed by local bitline of each cell (E_{lbl}) is derived

$$E_2 = E_{pre} + E_{ren} + E_{lwl} + E_{lsa} + 8 \cdot E_{lbl} \tag{14}$$

$$E_{lbl} = \left(E_2 - E_1\right)/7 \tag{15}$$

For further separating the elementary energy from the periphery circuits, a row unit circuit template is designed as shown in Fig. 13. In the same manner this fraction is separated by calculating E_3 and E_2.

$$E_3 = 8 \cdot \left(E_{pre} + E_{ren} + E_{lbl}\right) + E_{lwl} + E_{lsa} \tag{16}$$

$$E_{pre} + E_{ren} + E_{lwl} + E_{lsa} = \left(E_3 - E_2\right)/7 . \tag{17}$$

Similarly, a column circuit template is created in Fig. 14, by which the elementary energy consumed by vertical global bitlines (E_{vgbl}) is separated.

$$E_4 = E_{pre} + E_{ren} + E_{lwl} + E_{lsa} + 8 \cdot E_{lbl} + 7 \cdot E_{vgbl} \tag{18}$$

$$E_{vgbl} = \left(E_4 - E_2\right)/7 . \tag{19}$$

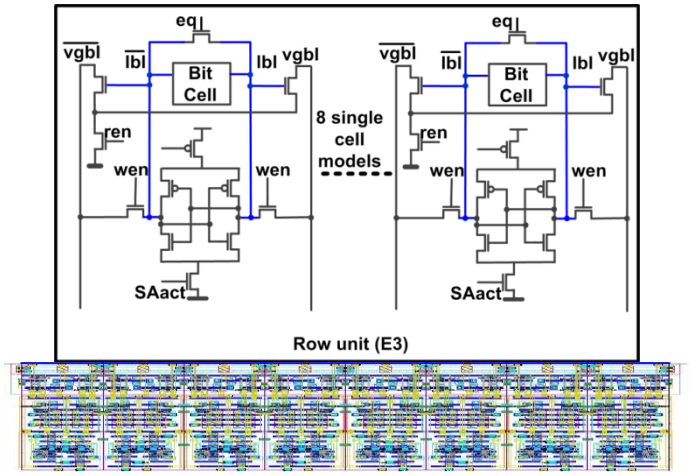

Fig. 13. A row unit circuit template

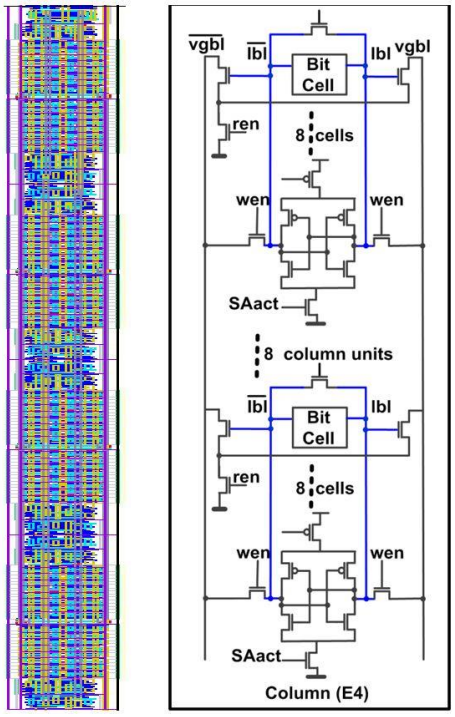

Fig. 14. A column circuit template

Since $E_1 \ldots E_4$ and $P_{static1} \ldots P_{static4}$ are pre-characterized by simulating the extracted-netlists of the four circuit templates, the elementary energy values E_{lbl}, $E_{pre+ren+lwl+lsa}$, E_{vgbl} can be derived. This way the dynamic energy for read/write operations and static power of the four circuit templates are obtained. It is assumed that a toggle condition occurs for each write operation. As before, the simulation configuration is TT corner, 25°C, 400MHz and 0.9V supply voltage in 40nm CMOS technology. The voltage swing of *vgbl* pair was chosen to 300mV to guarantee robust operations. The estimation approach is the same for other technology corners but the pre-characterization must be modified based on a Monte-Carlo simulation.

Table 4. Energy of four circuit templates (TT, 25 °, 400MHz, V_{DDH}=0.9V, V_{DDL}=0.3V)

Circuit Templates		Cell	Column unit	Row unit	Column
Dynamic Energy(pJ)	Write	4.26	5.25	25.75	12.58
	Read	2.91	3.71	22.47	5.88
Static Power (nW)		3.23	24.37	25.88	195.74

Read operation. As mentioned before, read and write operations are studied separately due to their different characteristics. For a read operation of a hierarchical SRAM with the use of LSAs, the bitline/wordline capacitance, the LSA along with read/write assist and precharge circuit are the main energy consuming components. The dynamic energies E_{lbl} and E_{lwl} are the sum of the energies due to the wiring capacitance itself and the capacitances attached to the memory cells. In addition, the energy consumed by the static components of the unselected memory matrices is attributed to the dynamic energy, comprising a significant portion. The static power ($P_{static1}$…$P_{static4}$) of the four circuit templates can be acquired using the same approach as before by separating the dynamic energy of each component. Particularly the pass transistors acting as column selectors are also included in the model. The static power of the global sense amplifiers (GSA) and the pass transistors are obtained by multiplying their count with the static power of two simple circuits: a GSA circuit and a pass transistor circuit. As a consequence, according to the parameters of memory matrix defined above for a hierarchical architecture, standby power of a column block can be estimated as

$$\begin{aligned} P_{static\_col} &= w\cdot\left(U\cdot N\cdot P_{lbl\_static}+P_{gsa\_pass\_static}\right)+w\cdot\left(P_{pre\_static}+P_{ren\_static}+P_{lwl\_static}+P_{lsa\_static}\right) \\ &= w\cdot\left(U\cdot N\cdot\left(P_{static\,2}-P_{static\,1}\right)/7+P_{gsa\_pass\_static}\right)+w\cdot\left(\left(P_{static\,3}-P_{static\,2}\right)/7\right) \end{aligned} \tag{20}$$

Finally, the overall dynamic energy of reading a bit from the memory matrix is estimated. The partitioning parameters (m, n, u) are converted to the number ($M=2^m$, $N=2^n$, $U=2^u$) of partitioned components in memory matrix. The total energy is calculated by parameterized accumulating the elementary energies (E_{lbl}, $E_{pre+ren+lwl+lsa}$, E_{vgbl}, E_{gsa}, E_{pass}). Particularly, the energy from the unselected parts in the memory matrices is calculated independently and then added to the total dynamic energy.

$$\begin{aligned} E_{read\_bit} &= w\cdot U\cdot E_{lbl}+E_{vgbl}\cdot\left(N-1\right)+E_{gsa}+M\cdot w\cdot E_{pass}+ \\ &\left(E_{pre}+E_{ren}+E_{lwl}+E_{lsa}\right)+(M-1)\cdot P_{static\_col}/f \\ &= w\cdot U\cdot\left(E_2-E_1\right)/7+\left(N-1\right)\cdot\left(E_4-E_2\right)/7+E_{gsa}+M\cdot w\cdot E_{pass}+ \end{aligned}$$

$$(E_3 - E_2)/7 + (M-1)P_{static\_col} / f \tag{21}$$

Write operation. For the write operation, the method to separate the dynamic write energy of each component is similar but using the pre-characterized write energies in Table 4. Particularly, the toggle state is not considered here because the energy of the write operation is obtained assuming a toggle event for each write. A toggle state does not occur all the time and its corresponding energy can be estimated using a similar approach as in [2] using a toggling probability. As a result, the write cycle energy can be calculated as follows.

$$\begin{aligned} E_{write\_bit} &= w \cdot U \cdot E'_{lbl} + \left(E'_{pre} + E'_{wen} + E'_{lwl} + E'_{lsa}\right) + \\ &\left(N-1\right) \cdot E'_{vgbl} + (M-1)P_{standby\_col} / f \\ &= w \cdot U \cdot \left(E'_2 - E'_1\right)/7 + \left(E'_3 - E'_2\right)/7 + \\ &\left(N-1\right) \cdot \left(E'_4 - E'_2\right)/7 + (M-1) \cdot P_{standby\_col} / f \end{aligned} \tag{22}$$

5.2 Verification of memory matrix model

Several memory matrices have been simulated in 40-nm CMOS technology to validate the model equations above. In Fig. 15 the dynamic power break-down of a 64-KByte memory matrix is shown. It is observed that the energy of local bitlines dominates the power consumption in a memory matrix as compared to the other circuits.

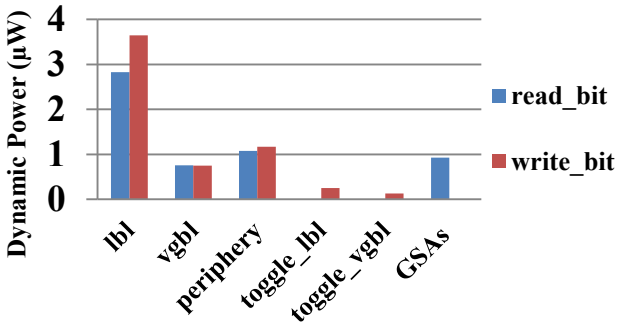

Fig. 15. Dynamic power component for a memory matrix (64 words 8 bit)

Fig. 16 shows a comparison of simulation and estimation data for four memory matrices of different capacities (64, 128, 256, 1K). Assuming a read access operation the four extracted-netlists were simulated using the same configuration. As shown in Table 5, the dynamic energies are compared to the estimated data and the differences

are below 10%. For the leakage power, the same comparisons are performed and the estimation errors are also below 10%.

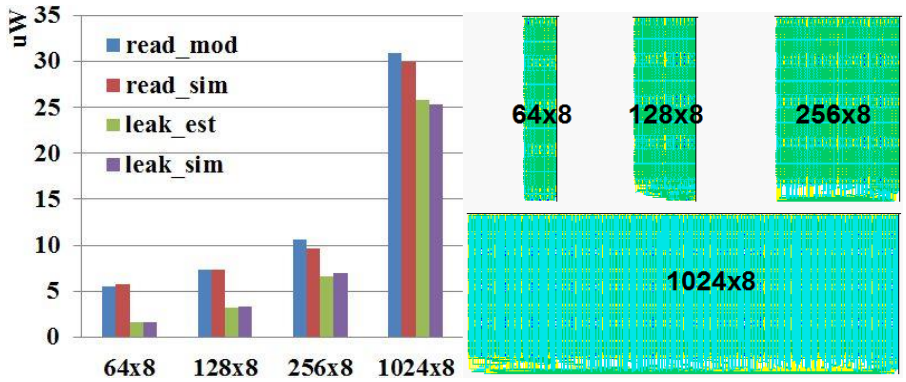

Fig. 16. Estimation and simulation data comparison for memory matrices with four capacities

Table 5. Model Estimation Errors of the four capacities (TT, 25 °, 400MHz, V_{DDH}=0.9V, V_{DDL}=0.3V)

Capacity	64 x 8bit	128 x 8bit	256 x 8bit	1024 x 8bit
Dynamic Energy	-5%	-1%	9%	3%
Static Power	-4%	-4%	-5%	2%

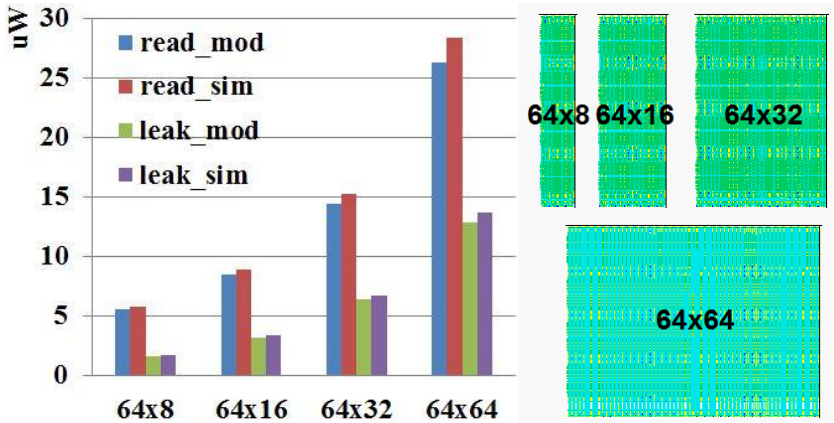

Fig. 17. Estimation and simulation data comparison for memory matrices with four word-lengths

Table 6. Model Estimation Errors of the four word lengths (TT, 25 °, 400MHz, V_{DDH}=0.9V, V_{DDL}=0.3V)

Word lengths	64 x 8bit	64 x 16bit	64 x 32bit	64 x 64bit
Dynamic Energy	-5%	-5%	-5%	-7%
Static Power	-4%	-4%	-3%	-6%

To further demonstrate the accuracy, four memory matrices with a fixed number of 64 words and with word lengths 8, 16, 32, 64 are implemented. As shown in Fig. 17 the estimation data are comparable to extracted-netlist simulation data. Both for dynamic energy and leakage power the estimation error remains below 10%, as listed in Table 6.

6 Optimization results

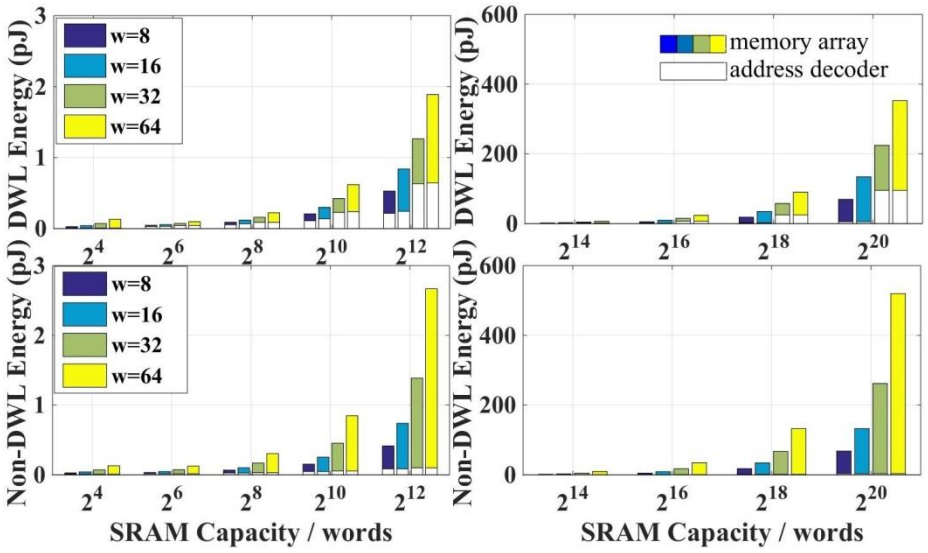

Fig. 18. Dynamic energy vs address bits & wordlengths for two architectures. The yellow bottom and the rest part of one bar represent the contributions of the address decoder and the memory matrix respectively.

The power model created in this work takes all the dominating power contributors of on-chip SRAMs into account, which include an address decoder, memory cells and assists circuits, local and global sense amplifiers, driver circuits and interconnect capacitance. Fig. 18 shows the power model applied for estimating the power consumption of SRAMs for various capacities and wordlengths. Minimum read dynamic power data is given due to more frequent read operation in caches. The model is applicable for capacities ranging from 16 to 1M words and four wordlengths (8, 16, 32, 64). The figure illustrates how DWL and hierarchical LSA architecture affect the read

power of SRAMs as function of different capacities and wordlengths. Moreover it indicates the different power contributions from address decoder and memory matrix to the dynamic read power.

In addition, the power model can be used for optimizing a specific SRAM by determining the optimal parameter combination. As discussed before, many possibilities exist for partitioning the memory matrix, the corresponding address decoder, given the three parameters partitioning parameters (m, n, u), and many options for circuit implementations. Depending on the optimization criteria parameter combinations are picked from all possible implementations options. Note that the impact of process variations on leakage power is included in the power model.

A Pareto-optimization is made by considering silicon area and read power of the different partitioning parameters. Fig. 19 shows how to use this approach to optimize a 1K Byte SRAM for achieving a power and area tradeoff. In the scatter plot, there are ten architectures presenting relatively good area and power, which are picked from all the generated architectures. Four Pareto-optimal implementations are marked, in which two architectures deliver low area and the other two deliver low read power. Depending on the user's requirements a selection can be made, for instance the green point deliving a favorable area/power tradoff.

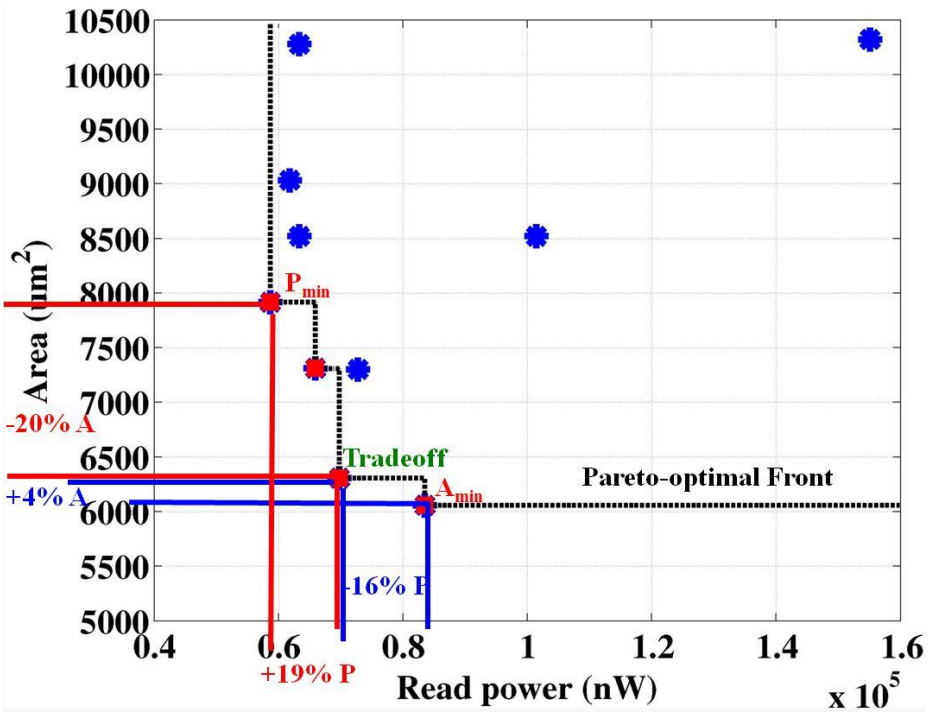

Fig. 19. Area cost vs read power tradeoff for a 1K-Byte SRAM

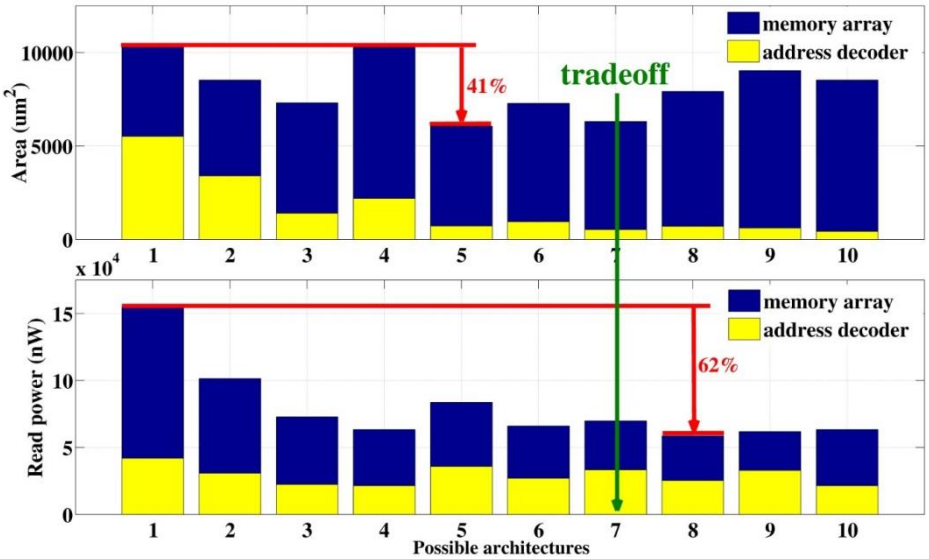

Fig. 20. Contribution of address decoder and memory matrix to area cost and read power for the 10 possible architectures of a 1K Byte SRAM

Fig. 20 shows a direct area/power break-down for the same ten possible architectures in Fig. 19. The contributions of the address decoder and memory matrix to the overall area and read power are shown and analyzed quantitatively. Between the worst case and best case solution a difference is observed for area and power up to 41% and 62% respectively.

7 Conclusion

In this chapter, a new method for power optimization of on-chip SRAMs comprising a hierarchical architecture was described. The method is based on a power model including various energy-efficient circuits and techniques. The introduction of the probabilistic estimation approach and the use of circuit templates provide quantified switching activities and pre-characterized customized circuits separately. Simultaneously the hierarchical architecture regarding many partitioning choices is defined by the partitioning parameters. The power model is verified by a variety of extracted-netlist simulations and it consistently exhibits good accuracy.

As a quantitative parameter optimization tool, this approach allows a fast and accurate power estimation of SRAMs comprising various capacities and wordlengths. In a hierarchical-architecture SRAM, the impact of partitioning with circuit selections on power and area were evaluated. The optimal architecture and circuits can be identified very quickly and accurately which leads to a SRAM specification with an achievable and attractive power consumption and silicon area. Moreover, this approach allows an easy tradeoff between area and power for meeting different design requirements. Furthermore, the power model can also be employed as a customized benchmark for

comparing various local circuits using the same architecture. Finally, this approach can easily be extended to other CMOS technologies due to its circuit templates and switching activity analysis.

References

1. Sharma, V., et al. (2011). A 4.4 pJ/access 80 MHz, 128 kbit variability resilient SRAM with Multi-Sized sense amplifier redundancy. Solid-State Circuits, IEEE Journal of 46 (10): 2416-2430.
2. Clerc, S., et al. (2012). A 65nm SRAM achieving 250mV retention and 350mV, 1MHz, 55fJ/bit access energy, with bit-interleaved radiation soft error tolerance. In: ESSCIRC (ESSCIRC), 2012 Proceedings of the. IEEE, pp. 313-316.
3. Rooseleer, B. and Dehaene, W. (2013). A 40 nm, 454MHz 114 fJ/bit area-efficient SRAM memory with integrated charge pump. In: ESSCIRC (ESSCIRC), 2013 Proceedings of the. IEEE, pp. 201-204.
4. Ren, Y. and Noll, T. G. (2013). An accurate power estimation model for low-power hierarchical-architecture SRAMs. In: Very Large Scale Integration (VLSI-SoC), 2013 IFIP/IEEE 21st International Conference on. IEEE, pp. 144-149.
5. Ren, Y., et al. (2012). Low power 6T-SRAM with tree address decoder using a new equalizer precharge scheme. In: SOC Conference (SOCC), 2012 IEEE International. IEEE, pp. 224-229.
6. Muralimanohar, N., et al. (2009). CACTI 6.0: A tool to model large caches. Tech. rep. http://www.hpl.hp.com/techreports/2009/HPL-2009-85.pdf.
7. Liang, X., et al. (2007). Architectural power models for sram and cam structures based on hybrid analytical/empirical techniques. In: Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on. IEEE, pp. 824-830.
8. Do, M. Q., et al. (2007). Leakage-Conscious Architecture-Level power estimation for partitioned and Power-Gated SRAM arrays. In: Quality Electronic Design, 2007. ISQED '07. 8th International Symposium on. IEEE, Washington, DC, USA, pp. 185-191.
9. Donkoh, E., et al. (2012). A hybrid and adaptive model for predicting register file and SRAM power using a reference design. In: Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE. IEEE, pp. 62-67.
10. Sun, L., et al. (2013). Low power and robust binary tree SRAM design for embedded systems. In: Electronic System Design (ISED), 2013 International Symposium on. IEEE, pp. 87-92.