ADAPTIVE CMOS 8T SRAM OPTIMIZATION USING REINFORCEMENT LEARNING FOR EFFICIENT POWER, DELAY, AND AREA TRADE-OFFS

[1] DONDAPATI YOGAIAH, [2] P. SURYANARAYANA, M.Tech.
[1] PG SCHOLAR, IN ECE Dept. ELURU COLLEGE OF ENGINEERING AND TECHNOLOGY, ELURU.
dondapatiyogi26@gmail.com
[2] ASSISTANT PROFESSOR, IN ECE Dept. ELURU COLLEGE OF ENGINEERING AND TECHNOLOGY, ELURU. Suryapothunuri@gmail.com

ABSTRACT: We build and optimize an 8T-SRAM cell in 18 nm CMOS technology that utilizes Reinforcement Learning (RL) to achieve tunable power, latency and area trade-offs in this study. For modern System-on-Chip (SoC) architectures, the SRAM is a critical component, and optimum performance of low-power, high-speed VLSI applications is crucial. However, in a scenario where multiple competing goals exist, a "best" design point may not be obvious with the traditional manual transistor scaling process.

This is overcome by the suggested method that is optimising the 8-transistor SRAM bit-cell architecture using Reinforcement Learning. In this method the RL agent self-learns features of the memory cells to find the optimal characteristics in combination with the sizes of the transistors and the operational conditions. The 8T SRAM design was selected over the more traditional 6T SRAM cells, due to its low voltage operation, read disturbance reduction, and better compatibility with NoC.The 8T SRAM design was selected, because of its low voltage operation, read disturbance reduction, and compatibility with NoC. read stability. The optimisation method reduces the averaged power consumption, propagation latency and cell area while maintaining the reliable read/write/hold.

The circuit diagram, discharge analysis and performance evaluation of the entire 18 nm CMOS-based SRAM cell is realized using Tanner EDA..

INDEX TERMS: Optimisation for power-delay-area (PDA), memory design, 8T-SRAM, CMOS, low power electronics and optimisation for analogue circuits. Optimization with a Pareto front, Multiple objective optimizations.

1. INTRODUCTION

Static Random Access Memory (SRAM) is a vital memory component in modern digital systems and is widely used in the electronic components such as

microprocessor, embedding controller, cache memory, AI accelerators, and IoT devices. Due to its high speed of read and write, low latency, and ability to be accessed by CMOS logic circuits, SRAM has become the most popular on-chip memory type used in very large scale integration (VLSI) devices. In modern semiconductor technology, SRAM arrays take up a lot of space on the device and use a lot of power overall. Consequently, exploitation of SRAM is one of the main objectives of modern design of integrated circuits with respect to power consumption, latency, stability and silicon area.

As CMOS technology advances to the deep a submicron and nanometre scales, SRAM design is strongly challenged with many important hurdles. The extremely small size of 18 nm technology transistors introduces a variety of concerns, including enhanced short-channel effects, leaking currents, manufacturing variations, lower noise margins, and reliable concerns. Because of these impacts, traditional SRAM cells perform worse and memory

The size of the transistors is growing more and more important and the operating conditions more critical for the architecture. In particular, the traditional 6T SRAM circuit suffers from problems such as read disturb, poor performance when operating the device at low supply voltages, and unequal read and write currents. These limitations have led to the development of a number of alternative SRAM designs, such as the 8T (eight transistor) SRAM.These restrictions have also generated a great deal of interest in other SRAM designs, including the 8T (eight transistor) SRAM.

When compared to the more traditional 6T SRAM design, the 8T SRAM cells has many significant benefits. The 8T design's separate read and write paths provide advantages for enhanced data stability and decreased risk of data corruption during read and write operations to the memory. The cell is better suited for low-voltage operation and its static noise margin (SNM) is increased since storage nodes are segregated from the reading bit- line during reading operation thanks to this separation. Although the 8T SRAM cell provides excellent functionality and reliability, novel challenges arise in designing the cell, particularly in terms of transistor scaling, leakage control, delay optimisation, and space overhead. Making the best compromise between these factors is difficult, as increasing the value of one indicator may decrease the value of another. The wider transistors can improve write speed and area, but they'll also consume more power and generate more heat.

The optimization of the SRAM cells has involved, in the past, manual device size variations, variable sweeps, and tuning done based on multiple simulations. In the

case of multiple design objectives, however, these conventional methods can be time-consuming, expensive to compute and even dependent on the level of the designer's expertise. Manual optimisation is becoming less and less effective due to the design space's size and non-linearity at advanced technological nodes such as 18 nm. This makes smart automated optimisation algorithms that can determine the optimal design parameters in an easy and quick manner without depending on cumbersome trial-and-error processes increasingly demanded.

This study proposes using the Tanner EDA tool in 18 nm a CMOS to develop and optimise a RL-powered CMOS 8T SRAM cell. Tanner EDA's right platform can be used to take care of nanoscale memory circuit layout, transient simulation, and functionality assessment. An adaptable and intelligent design framework is created by combining algorithm-level reinforcement learning optimisation with circuit-level SRAM design in the suggested approach. The iterative interaction with the SRAM simulated environment and the evaluation of their performance by the RL agent results in improved design quality.

Better circuit performance, and selection of more appropriate parameters of the transistor size.

Read, write and hold operations of the cells should be reliable, and in this study 8T SRAM cells are optimised, with adjustable and variable trade-offs between silicon area, propagation latency, and power consumption. The proposed approach aims to overcome the limitations of conventional SRAM design techniques and to help develop next-generation low-power or high-performance memory systems, using the brains of Reinforcement Learning to bolster the robustness of the 8T SRAM design.

Overall, this paper is a significant step towards the integration of AI aided design techniques in VLISC memory circuits resulting in better automated, adaptive & application specific optimisation of SRAM in future semiconductors.

2. LITERATURE SURVEY

The 8T-SRAM cell was extensively studied as an alternative to the low-power, good read stability 6T cell. The separation of the read and write channels as was suggested by Chang et al. [1] eliminates the read disturbance failures, which are a significant problem for 6T cells well beyond sub-65 nm. Regarding 8T read power and create a writing margin, Verma & Chandrakasan [2] developed analytical models, which gave

designers closed-form formulae to use when sizing transistors initially. However, in deeply scaled technologies, these models will have incorrect assumptions, such as long channel behaviour and ignoring velocity saturation.

The first step is typically to sizewrite-access transistors to meet the write margin; the second is to design the read-port transistor to meet the specified read current (speed); and the third is to adjust the leakage by choosing the threshold voltage. This method may sound basic, but it doesn't take into consideration the complex interactions between transistors. For example, increasing the pull-down transistor size of the read port improves the read current (read latency) but will increase the bitline capacitance, thus increasing the dynamic power. Increasing the size of write-access transistors enhances write- ability at the expense of half-select stability under similar circumstances. Although Takeda et al. [3] and even Guo et al. [4] have characterized all of these trade-offs, there is no systematic method to optimize simultaneously all competing goals. After discovering the limitations of manual design, researchers have optimized SRAM with different kinds of metaheuristic algorithms.However, due to limitations of manual design, researchers have optimized SRAM with different kinds of metaheuristic algorithms. To optimize the size of 6T-SRAM cells for low current leakage during writing and reading

stability requirements, Sani et al. [5] used a genetic algorithm (GA). Although their method required over 5,000 SPICE simulations for each optimisation run, it reduced leakage by 15% when compared to designs that were scaled manually. Similarly, Jafari et al. [6] optimized an 8T-SRAM based on particle swarm optimisation (PSO), considering read latency, dynamic power consumption in a weighted total. Both PSO and GA converged more quickly than GA, but they were unable to provide several trade-off solution in a single run and tended to converge too quickly to local optima.

In SRAM design, the Pareto fronts have been directly obtained using multi-objective evolutionary algorithms (MOEAs) such as NSGA-II [7] and SPEA2. Optimising 8T-SRAM for read latency and leakage power, Pal et al. [8] were able to select 20 designs using NSGA-II that were all non-dominated. MOEAs require large populations (typically 100-200 individuals) and generations, which results in thousands of circuit simulations, too much for high fidelity SPICE models to compute. Moreover, these algorithms are offline only, which means that the designer cannot alter their minds at the end of optimisation since the process does not run online, and has to repeat the optimisation process.

The second more recent idea for decreasing the cost of simulation is to use surrogate-assisted simulation.

evolutionary algorithms. Li et al. [9] managed to reduce 70% of the percentage of SPICE evaluations by using learned

Gaussian model processes to imitate the performance parameters of SRAM. When there is significant nonlinear behaviour in the design space, for example at the stability boundaries, surrogate model accuracy drops, and retraining is required if the design parameters are changed.

3. EXISTING METHOD

The circuit of the 8T-SRAM is displayed in Figure 1. The M4, M5, M6 and M7 were classified as read/write transistors for testing in the future. M1 and M3 were categorised were NMOS, and M0 & M2 as PMOS. Delay of the delay of switching or keeping voltage affects read and write operations of Q's, 8-transistor static random-access memory (8T-SRAM) has an additional Read Word Line (RWL) to overcome interference from the Bit Line (BL) and Bit Line Bar (BLB), stable Q and QB. It is optimized for write as there is no effect of read route on the write route. Variable tolerance of 8TSRAMs leads to better read stability and write-ability and reduces power and latency [8]. To provide strong writes at the typical 6T cost (one sizing decision affecting both reads/writes), the RL agent might use a larger read-stack to meet speed/leakage goals on RBL & the attached write-access device(s) on BL/BLB.

Independence offers more co-optimization flexibility across the performance areas with our RL-driven design as it allows local sensing/evaluation circuits to be optimized independently, or even writing/reading bit lines can share bit lines with different bit line groups.

In the past few decades, CMOS circuit optimisation has been an important area of research, where minimising power consumption, latency, and area have all been emphasized. Manual tweaking, heuristic optimisation and evolutionary algorithms are some commonly used techniques that have proven to be too limited in the complexity of modern SRAM designs, especially at advanced technology nodes [9]. In early of integrated circuits design, engineers would manually tune transistor designs to fulfil design requirements. This approach is inefficient, inaccurate and time-consuming for complex and high dimensional designs [10]. Manual techniques are challenging in modern high-performance SRAM systems because of the complexity of the design space. A simulated (SA) [11] optimisation, innovative genetic algorithms (GA) [12] or particle swarm optimisation (PSO) [13] is often used in the optimisation of integrated circuits. Such methods evaluate various configurations within an architectural structure in order to find solutions that have good performance. Unfortunately, they can't handle several conflicting objectives, rely on significant

They are evaluated in a long time and are very complex and large in the search space, converging to local minima. In SRAM optimisation, the ML algorithms

based on the RL approach have been gaining ground. RL also has some benefits over the older methods, namely that it can explore vast high dimensional design spaces autonomously. The RL algorithm could consider performance feedback and changes of the design environment to keep updating the transistor layouts. RL is able to effectively simultaneously optimise power, latency & area; and therefore outperforms heuristic methods for solving multi-objective optimisation problems. RL has been used in many design tasks within the field of integrated circuits for many years, such as size optimization, minimization of power consumption and logic synthesis. RL techniques can be superior to traditional optimisation techniques because they can handle high-dimensional design spaces and accommodate changing design specifications. Agents learn acceptable designs via simulated interactions with the environment in RL-based CMOS design. These methods improve the design process in several ways, such as by identifying better, more efficient designs more quickly [7, 10]. It appears in the design of SRAMs using RL to be a viable approach, especially for optimising multi- objectives. Unlike existing approaches, RL can interact with the real-time dynamic optimisation of parameters at the transistor level. One of the benefits of RL in SRAM design is that it can be implemented to meet a balance of power, latency, and area design criteria, all of which are in conflict with each other.

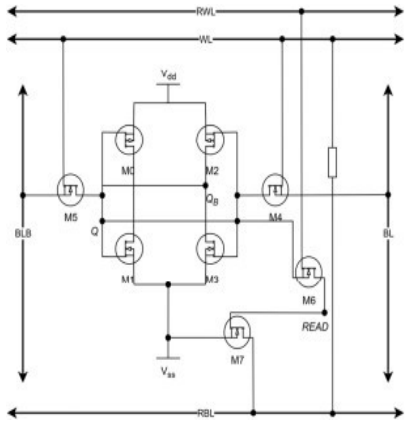

Fig 1. Circuit diagram for 8T-sram cell.

4. PROPOSED METHOD

In an 8T SRAM cell, two inverters (M2-M4 and M1-M3) are connected in a mutually exclusive manner. For single-ended read operation, two NMOS semiconductors (M7, M8) are used, and for single-ended write operation, one access transistor (M5, M6) is used, making it possible to implement the entire design [11]. The symbols WBLB, WBL and WWL represent the write word line creative bar, write bit line and write word line, respectively when performing a write operation. A read bit line (RBL) and a read phrase line (RWL) are used to accomplish the read operation. This cell can be operated as a read/write cell down to a voltage of

Operation in the sub-threshold region with 0.1V. The construction of 8T SRAM cell which has been suggested is shown in Fig. 2. The 8-transistor SRAM cell's read port (M7&M8) read aids readability

[12].connecting to the first inverter's gate terminal. When connected to a ground node a NMOS value turns into “zero”. By storing in the Q node, the M7 NMOS may be switched on or off, and reading the Bit Line (RBL) goes high when the Read Words Line (RWL) is enabled. The bit line stays high and M7 is switched off if the cell holds a value of 1. Figure 3 illustrates that the SRAM cell is set to "0" when it is discharged through the bit line M7 and M8. A static random access memory (SRAM) cell with 8 transistors has a single-ended read/ write terminal. The Writing Bit Lines (WBL) and Writing Word Line (WWL) are used for single ending write operations. To read, single ended operations are made use of with the help of reading Bit Lines (RBL) and Read Word Lines (RWL)..

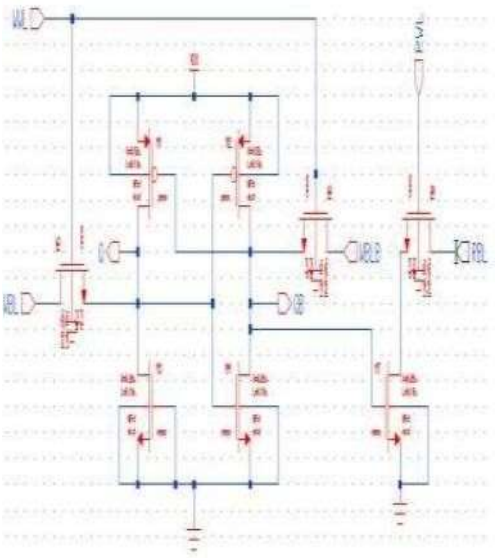

Fig. 2: Proposed 8T SRAM Using Extra Pass Transistors.

This technique bypasses the potential of any short circuit in the 8T SRAM cell by adding two extra pass transistors at either end of the cell. A single PMOS having a logic "0" at its gate terminal is placed between the cell and VDD as it is shown in figure 4. Also, a logic "1" cell with an NMOS gate is connected to ground via an NMOS device. Two pass transistors added to the 8T SRAM cell might help reduce the total power consumption..

5. SIMULATION RESULTS AND DISCUSSIONS

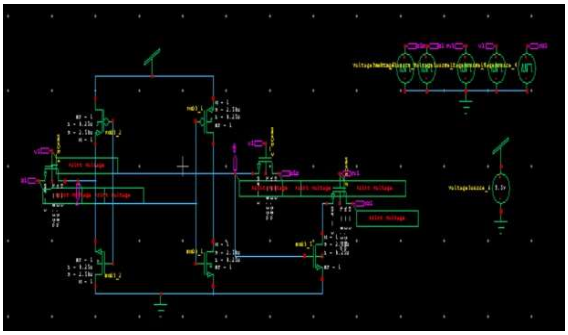

Fig 3. schematic of proposed 8T SRAM

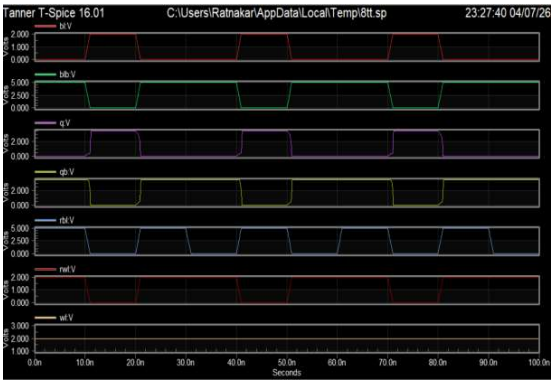

Fig 4. Simulation wave form of proposed 8T SRAM.

Table-1: delay analysis of existing and proposed methods

Applied Voltage	Standard 6T SRAM cell	Conventional 8T SRAM	Proposed 8T SRAM with Read assist	Proposed SRAM Extra P Transist
0.1V	273.78 PS	172.84 PS	49.591 PS	155.68

Table-2: power analysis of existing and proposed methods

Applied Voltage	Standard 6T SRAM cell	Conventional 8T SRAM	Proposed 8T SRAM with Read assist	P 8 u t T
0.1V	5.6280 µW	5.3201 µW	3.6867 µW	3

6. CONCLUSION & FUTURE SCOPE

The proposed implementation of 8T SRAM cell has lower latency and is biased at 0.1V which is suitable for ultra-low power applications. The two changes to the standard 8T SRAM cell are intended to improve the power and delay performance. A first 8- transistor system using a read-assist approach and a second 8-transistor system using a read-pass transistor. These are compared to standard 8T SRAM cells as well as the standard 6T SRAM cells. By using read assist, the suggested solution 8T SRAM with delay is able to decrease latency by 81.89% when compared to 6T SRAM cells and by 71.36% when juxtaposed with conventional 8T SRAM cells. The proposed 8T SRAM with two extra pass transistors has 43.14% and 9.93% less latency when compared to 6T SRAM cells and conventional 8T SRAM cells, respectively. The proposed 8T Read-aid SRAM achieves a power saving of 34.5 percent and 30.7 percent when compared to 6T SRAM and conventional 8T SRAM cell, respectively. Compared to the 6T SRAM cells and conventional 8T SRAM cells, the proposed 8T SRAM cell with two additional pass transistors consumes 44.42% and 41.20% less power, respectively. These circuits are validated with the ELDO simulation.

program. Using the Tanner EDA software program, the schematic circuits and layouts are developed.

FUTURE SCOPE: The present investigation aims to optimise a single 8T-SRAM cell. Expanding the RL structure to optimize all the memory subsystems, especially: Complete memory arrays must be optimized for the memory cell in addition to the sensing amplifiers, write drivers, address decoders and precharge circuits. If array level factors, such as size of columns/rows, bit line capacitance, which/sense amplifier offset etc., are included, a hierarchical RL design would have to be used where the different agents optimize different sub-blocks with a coordinated reward. Co-Design of Peripheral Circuits: The sensitivity of the sensing amplifier, the strength of the write driver, etc. of the peripheral circuits will decide the optimum cell size. An RL agent that optimizes peripheral parameters and cell size simultaneously may yield better results on system-level power-delay-area optimisation compared to sequential optimisation.

REFERENCES

[1] L. Chang et al., "An 8T-SRAM for variability tolerance and low-voltage operation in high-performance caches," *IEEE JSSC*, 2008.

[2] N. Verma and A. P. Chandrakasan, "A 256 kb 65 nm 8T subthreshold SRAM employing sense-amplifier redundancy," *IEEE JSSC*, 2008.

[3] K. Takeda et al., "A read-static-noise-margin-free SRAM cell for low-VDD and high-speed applications," *IEEE JSSC*, 2006.

[4] Z. Guo et al., "Large-scale SRAM variability characterization in 45 nm CMOS," *IEEE JSSC*, 2009.

[5] M. Sani et al., "Genetic algorithm based leakage current reduction for 6T SRAM cells," *Integration, the VLSI Journal*, 2015.

[6] A. Jafari et al., "PSO-based optimization of 8T SRAM cells for low-power applications," *Microelectronics Journal*, 2017.

[7] K. Deb et al., "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE TEC*, 2002.

[8] S. Pal et al., "Multi-objective optimization of 8T SRAM using NSGA-II," *IEEE ISQED*, 2019.

[9] J. Li et al., "Surrogate-assisted evolutionary optimization for SRAM design," *ACM TODAES*, 2020.

[10] H. Wang et al., "Deep reinforcement learning for analog circuit sizing," *IEEE ISCAS*, 2019.

[11] R. Budhathoki et al., "Reinforcement learning for analog circuit optimization with constraint awareness," *IEEE TCAD*, 2021.

[12] S. Kim et al., "RL-based write-assist optimization for low-power SRAM," *IEEE A-SSCC*, 2020.

[13] Y. Zhao et al., "PPO-based SRAM cell sizing with reduced simulation cost," *IEEE ICCAD*, 2022.