EE113 Project：SRAM with layout improvement and instruction set

Group: Zhang Yifan & Li Zhenbang

I. Basic: SRAM

The design of our SRAM is composed of five main components: address decoder, wordline buffer, SRAM array, time controller for multiple enabling signal generator, column union(write driver, sense amplifier and pre charger) and output registers for output signal remaining.

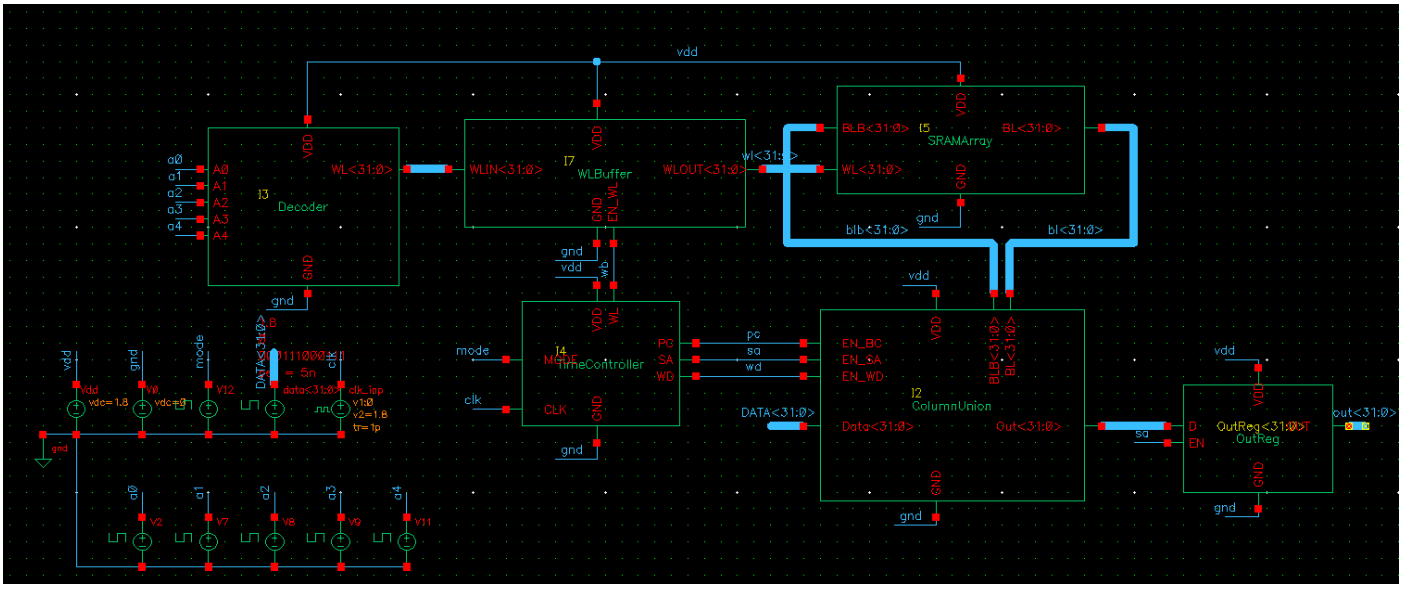


Fig.1. Basic check table

The most important time optimization method is logic effort computation and sizing optimization between the CMOS in different modules, especially the driven pulse generating part *(TimeController* in fig.1.) and the pulse receive part(sense amplifier, pre charger, write driver, wordline buffer).

However, this optimization should carefully deal with the buffer number and sizing way since that the pulse should be strong and wide enough to driven some follow up logics. Another important point is that the order of driven pulse should be ensured. In our adjustment, we find that the mineral overlap can be accepted but further will result in the disfunction of the circuit. Sometimes it is better to abandon some logic sizing in a few MOSFET levels and ensure a more stable medium output signal.

The area and speed are also a pair of trade-off element in sizing. However, in bonus2, we mainly discuss about the layout it self and several layout improvement strategies in regards of the area.

	Basic	Bonus2(layout area improvement)
Critical Path(ns)	1.34	\
Read Energy(fJ/b)	163.3	\
Write Energy(fJ/b)	246.8	\
Bit cell Array Area(m^2) (improved)	1.006E-08	
Total Area(m^2)	1.501E-08	\

Condition: smic18mmrf	our design	lithographically friendly 6T
1Kb Bitcelll Array Area(m^2)	1.006E-08	1.045E-08

(In bonus2, we mainly discuss on the area and different layout strategies about SRAM array.)

Fig.2. Basic check table

Explanation of Solutions to Checklist Requirements:

Ensuring the voltage of the discharged BL/BLB is no lower than 70% VDD in read mode:

Compared to write mode, the pulse on wordline which activates access NMOSs in SRAM cells is shorten so that the discharging time is limited, resulting in a voltage drop less than 30% VDD. We accomplished this by generate an independent pulse with another pulse. Logic effort computation and sizing is operated for the better delay outcome.

Bitcell width and length h causality:

For ensuring both read and write stability, the transistors must satisfy ratio constraints. The NMOS pulldown transistor in the cross-coupled inverters must be strongest. The PMOS pullup transistors must be the weakest. The access transistors are of intermediate strength. What's more to achieve good layout area outcome, all of the transistors must be relatively small. According to the lecture, our SRAM cell is sized as below:

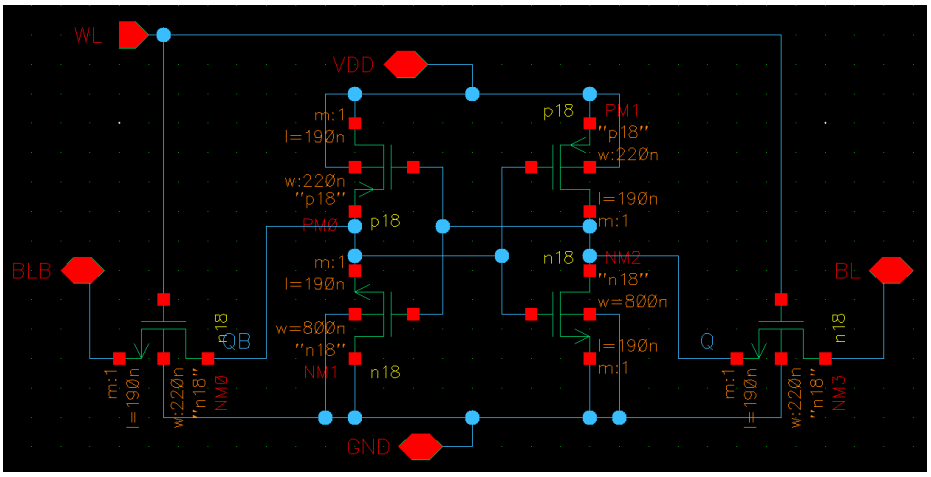


Fig.3. Cell sizing

II. Bonus1: SRAM with basic instruction set

In this part, the target is to make the SRAM and the appended D-flipflop array capable of doing five basic data transfer and arithmetic instructions. (DATA to SRAM, REG to SRAM, SRAM to REG, ADD, SHIFT) Since these instructions share some same bit patterns, they can share some same data paths, like register out, data mux and flipflop line write enable logics.

Compared to the SRAM in basic part, the instruction set realization requires a D-flipflop array for data restoring, ADD and shift operation, an instruction decoder, input and output mux in SRAM array for data picking and the Datapath with control logics and arithmetic operations.

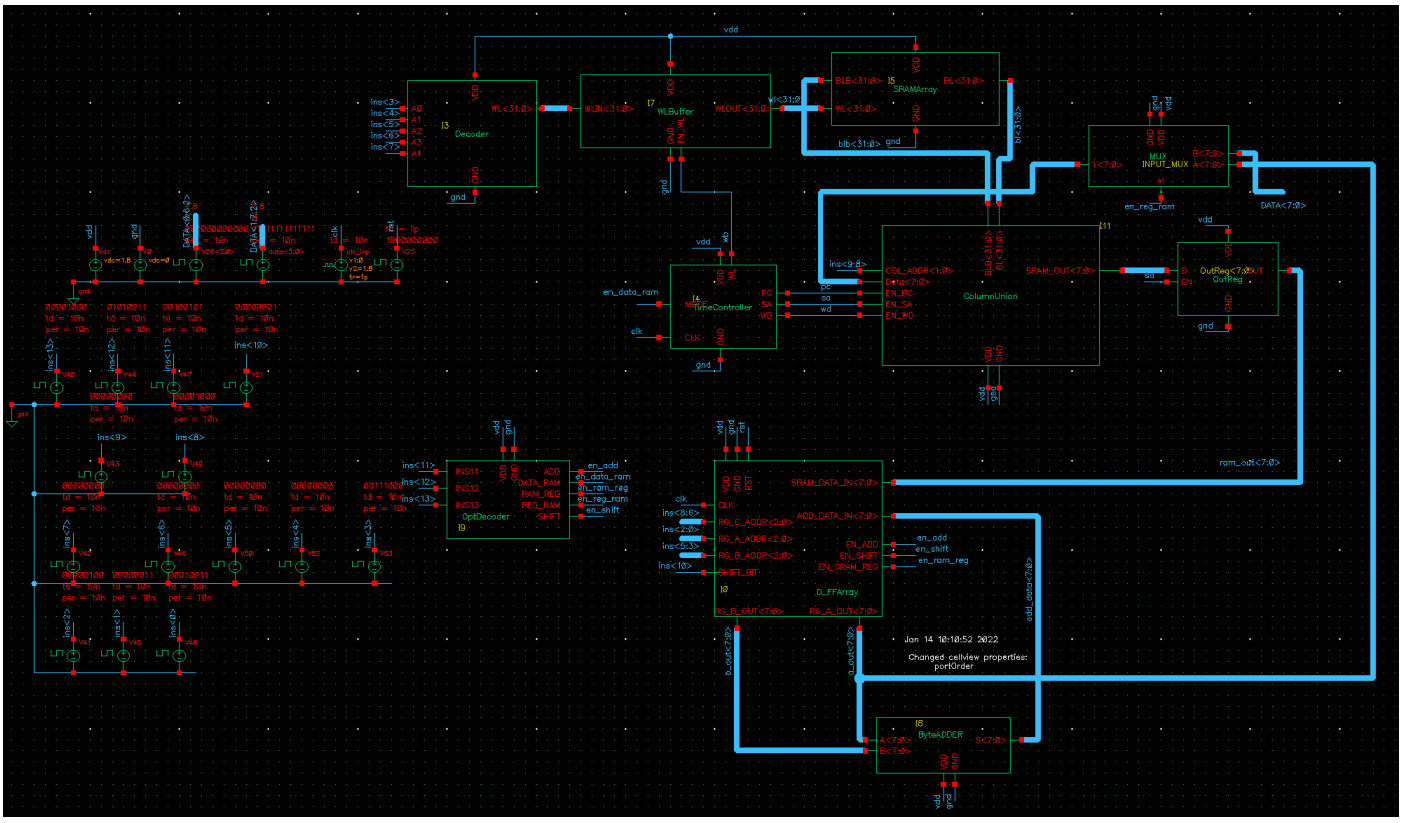


Fig.4. Whole schematic design: SRAM, flipflop array, Instruction decoder, Datapath and testbench

The core of design is the structure of D-flipflop cell and array, they involve in the four main instructions: REG to SRAM, SRAM to REG, ADD, SHIFT. Dealing the instruction sequence with correct data input and output in designative registers requires

a comprehensive input/output control logic for every instruction.

For the flipflop cell, we choose the TSPC structure with reset ports design to ensure the stability and avoid any possible float points. To realize the function of register write instructions (SRAM to DATA, ADD, SHIFT), a 2 level static cmos MUX logic is used, enabling the data from shift, add result or SRAM been correctly chosen and converted to TSPC.

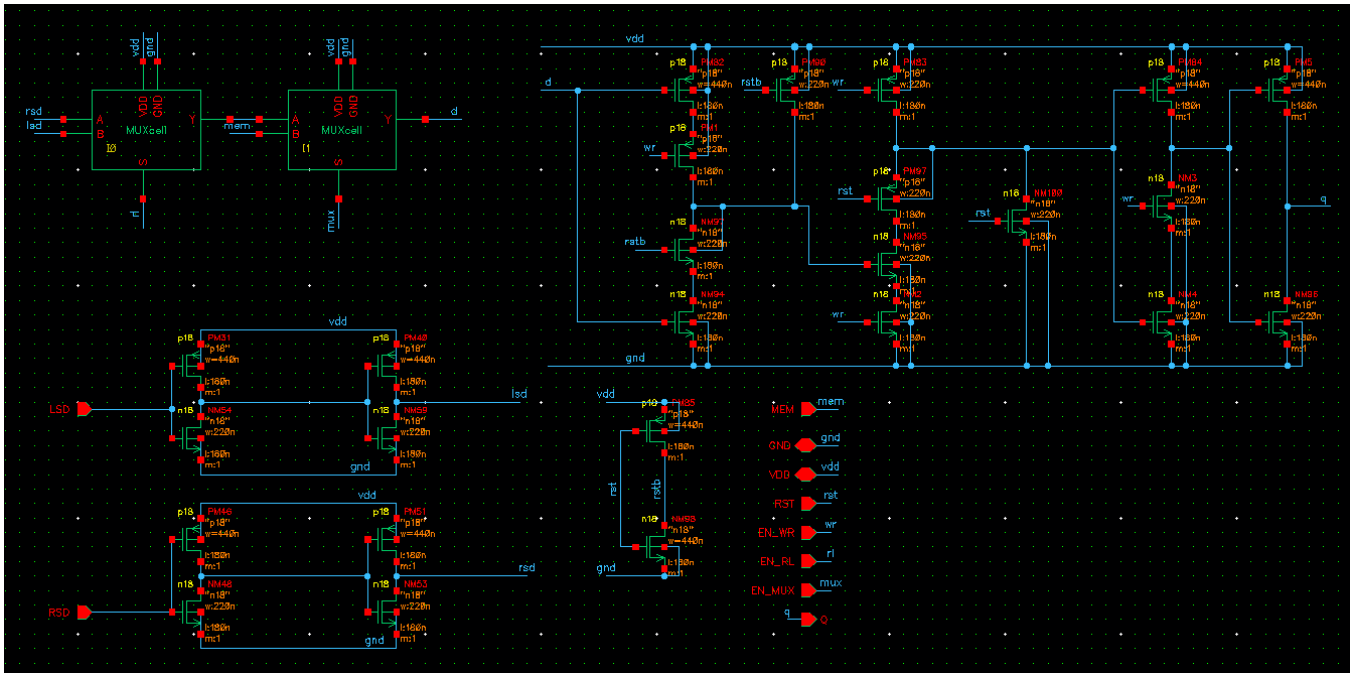
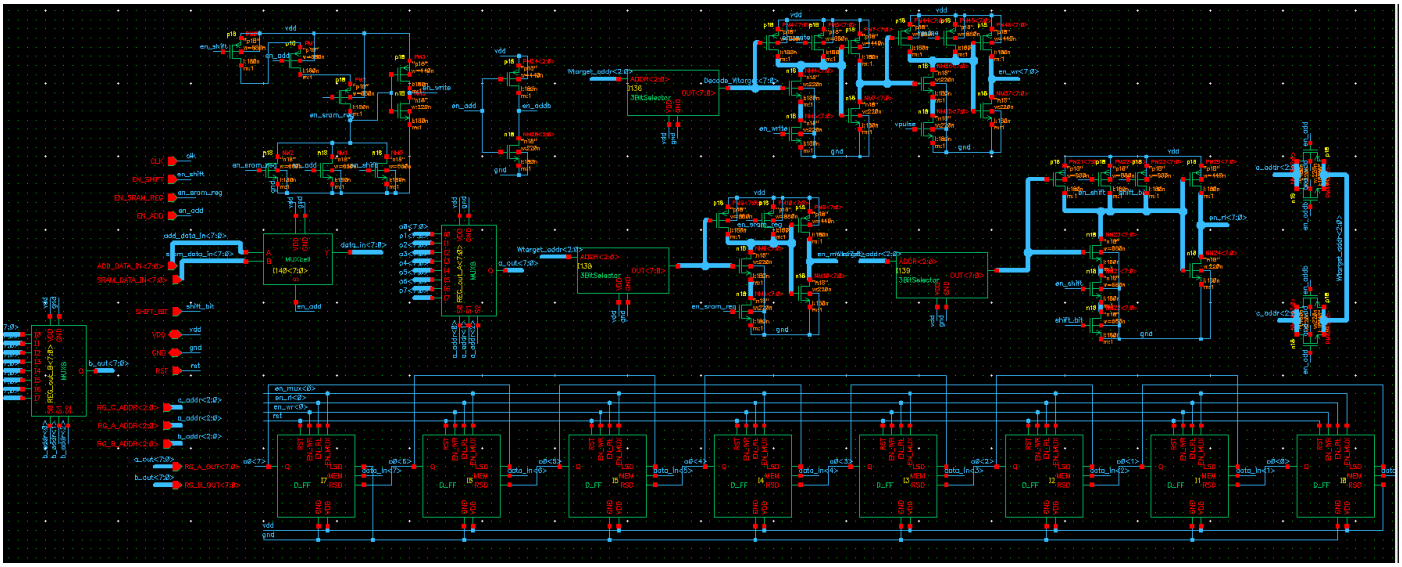


Fig.5. TSPC flipflop with reset and enable logic

For the D-flipflop array, a comprehensive static logic control is designed for the three write-related instructions(SRAM-REG, ADD, SHIFT), *en_write* sigs is decided and multiplexed with decoded register address and enable the distinct registers targeted. *Pulse* is generated, multiplexed with *en_write* transported to the flipflops. *en_mux* sigs and *en_rl* sigs decide the data in path between add result/shifted data and the shift direction.

For simplification, we reuse some single data paths for multiple instructions. The register out is used by both register out in normal read and register A out in add operation. The all three data writing instructions share the same inner path to flipflop lines. The main idea is put all the data multiplexing and address choosing work just at the port and make sure that the inner data path and control logic between flipflop cells and lines as simple as possible.



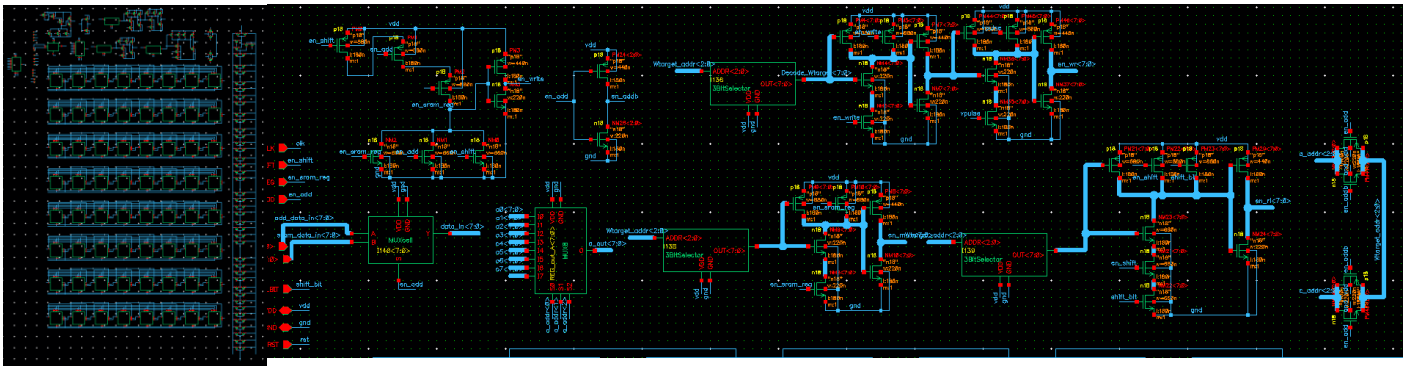


Fig.6. Flipflop array and its input/output control logic.

Register read instruction (REG to SRAM) special because its actually not related with D-flipflop array control signals but the SRAM array input control signals, the register output can always operating and be blocked by SRAM array input data multiplexer. This idea of constant output with input blocking between blocks is quite useful in all modules.

III. Bonus 2: Layout strategies and area analysis

In this part, we mainly improved the area cost of SRAM array, which is quite important in SOC design or embedded Systems, where the bitcell density of SRAM is considered as of high importance. We made some improvement according to the papers and the book *CMOS VLSI Design: A Circuits And Systems.*

For the SRAM array, the cell is designed to be mirrored and overlapped in order to share VDD and GND lines between adjacent cells all along the cell boundary. A single diffusion contact to the bitline is shared between a pair of cells. This halves the diffusion capacitance, and hence reduces the delay discharging the bitline during a read access and greatly improve the cell density.

The wordline is run in both metal1 and polysilicon. These two layers are occasionally be strapped every two or cells between.

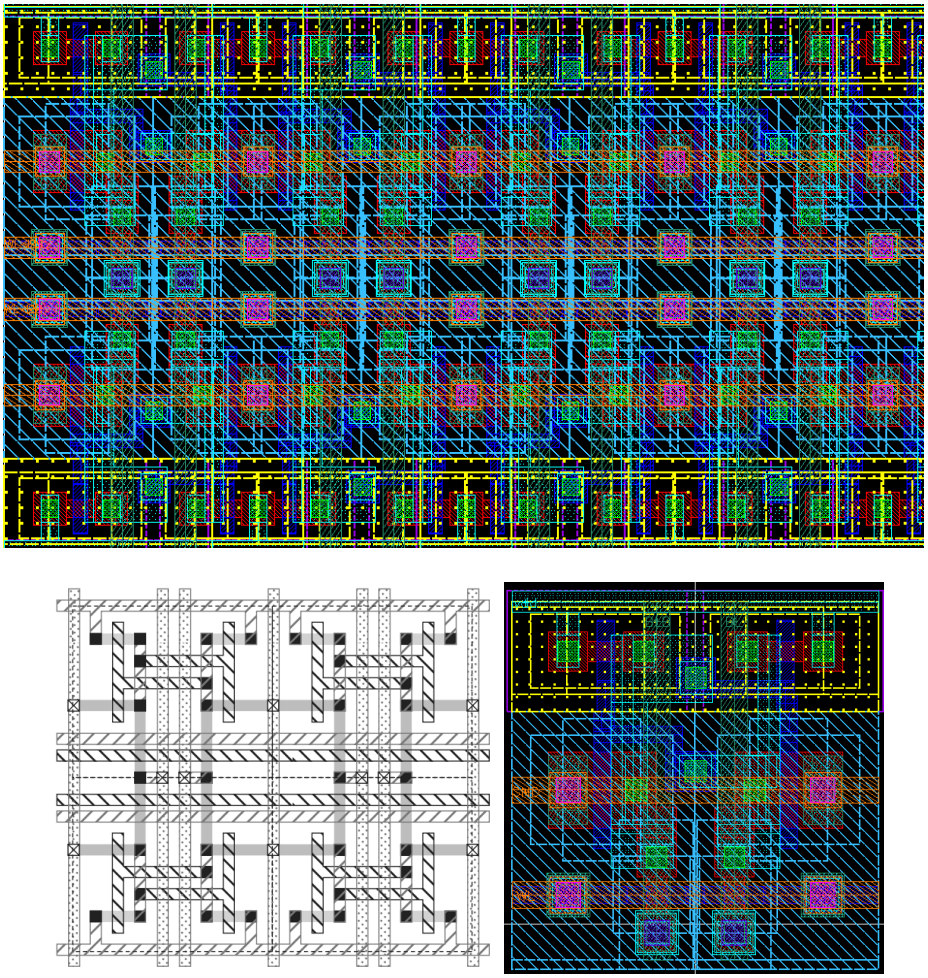


Fig.7. Cell layout and the adjacent cell arrangement

Comparison between different strategies of bitcell layout:

When its feature size is smaller than the wavelength of light, the bends in polysilicon and diffusion are difficult to precisely fabricate. Since then, today's nanometer processes widely use *the variability lithographically friendly 6T cell,* where diffusion runs strictly in the vertical direction and polysilicon runs strictly in the horizontal direction.

The core idea of this cell is reducing the critical bitline capacitance at the expense of longer wordlines. Its layout occupies two horizontal metal1 tracks and six vertical metal2 tracks and uses local interconnect or trench contacts to bridge between the PMOS drain and the NMOS transistors and polysilicon routing. Again, substrate and well contacts are shared between multiple cells like what we do in layout. However, this layout strategy can have a better density outcome compared with our's original design in scale in some special case.

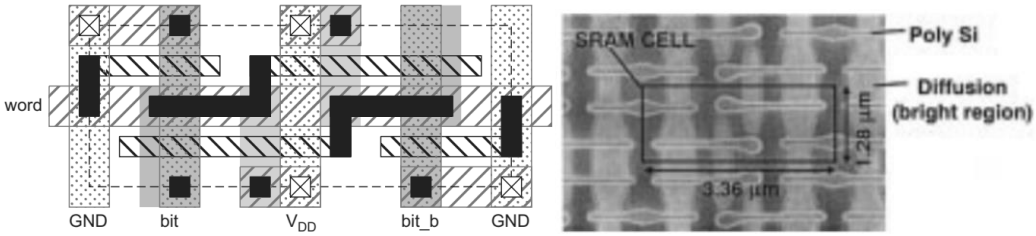


Fig.8. variability lithographically friendly 6T cell (pic source from reference [1] [4])

For such SRAM cell, which is 2.8um in length 6.5um in width, with share of diffusion regions and holes between adjacent cells, 1.6 um per cell is realized in an array, So the array size can be reduced to 6.5*1.6*32*32 without. The actual size is little bigger if edge effect is taken into consideration.

Condition: smic18mmrf	our design	lithographically friendly 6T
1Kb Bitcelll Array Area(m^2)	1.006E-08	1.045E-08

Fig.9. layout strategies performance comparison

REFERENCES

[1] Weste N, Harris D. Cmos Vlsi Design: A Circuits And Systems[J]. Perspective,. Pearson-Addison Wesley, 2010.

[2] C. A. Kumar, B. K. Madhavi and K. Lalkishore, "Performance analysis of low power 6T SRAM cell in 180nm and 90nm," 2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2016, pp. 351-357, doi: 10.1109/AEEICB.2016.7538307.

Papers from IEEE Journal of Solid-State Circuits (JSSC):

[3] M. -F. Chang et al., "A Compact-Area Low-VDDmin 6T SRAM With Improvement in Cell Stability, Read Speed, and Write Margin Using a Dual-Split-Control-Assist Scheme," in IEEE Journal of Solid-State Circuits, vol. 52, no. 9, pp. 2498-2514, Sept. 2017, doi: 10.1109/JSSC.2017.2701547.

[4] K. Osada et al., "Universal-V/sub dd/ 0.65-2.0-V 32-kB cache using a voltage-adapted timing-generation scheme and a lithographically symmetrical cell," in IEEE Journal of Solid-State Circuits, vol. 36, no. 11, pp. 1738-1744, Nov. 2001, doi: 10.1109/4.962296.