A Comparison of Full Swing and Partial Swing SRAM Read Topologies

by

Bao Gia Truong, B.S.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

August 2009

A Comparison of Full Swing and Partial Swing SRAM Read Topologies

Approved by
Supervising Committee:

Jacob Abraham, Supervisor

Mark McDermott

Dedicated to my wife, Joanne.

Acknowledgements

I would like to acknowledge the help of my family, professors, and colleagues. Their assistance and guidance was invaluable during the course of my education.

August 2009

A Comparison of Full Swing and Partial Swing SRAM Read Topologies

Bao Gia Truong, M.S.E.

The University of Texas at Austin, 2009

Supervisor: Jacob Abraham

This paper outlines design considerations and implementation details of full swing and of partial swing SRAM arrays. Comparisons between the two methods based on performance, power, and noise rejection are then presented. Finally, a decision matrix will be provided that selects the better topology based on varying design constraints.

Table of Contents

Chapter 1

Background Information

1.1 Introduction

The usage of SRAM (static random access memory) is prevalent in CMOS applications needing high density and high performance memory. SRAM cells are vastly smaller than latch memories and are therefore used extensively in processors where storage must reside close to execution units and take up as little space as possible. To achieve high density, SRAM memory cells push the boundaries of CMOS process to the smallest possible device dimensions. With small dimensions, device variation can have an immense impact on memory yield and performance. For reasons like this, various cache topologies exist to maximize different benefits. Whether it is performance, density, stability, writability, simplicity, or power efficiency, there is most likely an optimal solution for each.

This paper will give background SRAM information and go through the construction of two SRAM systems: one employing partial swing and the other full swing read. The partial swing SRAM introduces an analog component in the form of a sense amplifier meant to detect small voltage differences where the full swing design relies on a classic dynamic precharge and evaluate scheme. The circuit components and operations of both read methods will be discussed. In addition, SPICE simulation data evaluating the advantages and disadvantages of each will be presented. Comparisons will be made based on performance, leakage power, active power, and noise robustness to provide a decision making mechanism that will allow the designer to choose between full swing and partial swing SRAM's without significant design work.

1.2 6T Memory Cell

The six transistor cell (6T) schematic shown in Figure 1.1 is one of the most common memory cells used in SRAM arrays.

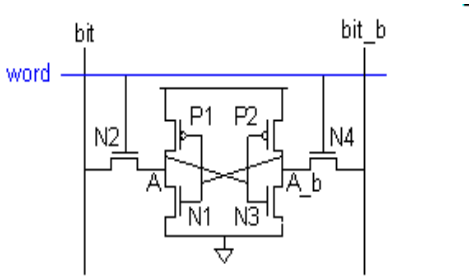


Figure 1.1: 6T schematic [8]

The sizes of the six devices in an SRAM cell are chosen to balance read performance, write performance, density, and stability. To perform a read, for example, the 6T cell only needs to discharge 'bit' or 'bit_b.' Therefore, read performance can be improved if N1, N2, N3, and N4 are increased in size. P1 and P2 only need to hold a bitline at the precharge voltage and are therefore small.

Figure 1.2 and 1.3 are scanning electron micrographs (SEM) of 45 nm 6T SRAM cells from Intel and IBM, respectively. Exact device dimensions are not available but devices N1 and N3 are clearly larger than P1, P2, N2, and N4 in both SEM's.

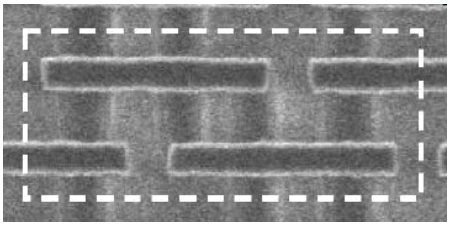


Figure 1.2: Intel 45 nm 6T cell [1]

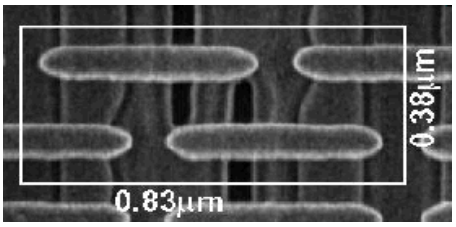


Figure 1.3: IBM 45 nm 6T cell [2]

1.3 SRAM Read and Write Operations

P1, P2, N1, and N3 from Figure 1.1 form cross coupled inverters that actively hold a static state. N2 and N4 are used to expose/isolate internal nodes ‘A’ and ‘A_b’ to/from the bitlines labeled 'bit' and 'bit_b.' As shown in Figure 1.4, 'bit' and 'bit_b' are first precharged to the supply voltage during a read operation. N2 and N4 are activated ('word' asserted) and either 'bit' or 'bit_b' is pulled low by N1 or N3 based on the state of the cross coupled inverters.

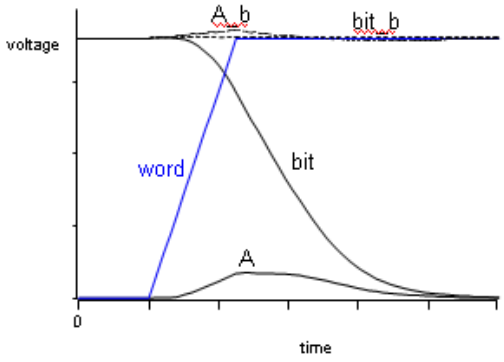


Figure 1.4: Full swing read operation [8]

In the write operation shown in Figure 1.5, 'bit' and 'bit_b' are driven to complementary voltages, N2 and N4 are activated, and the cross couple inverters assume a voltage at node ‘A’ that matches 'bit' and a voltage at ‘A_b’ that matches 'bit_b.’

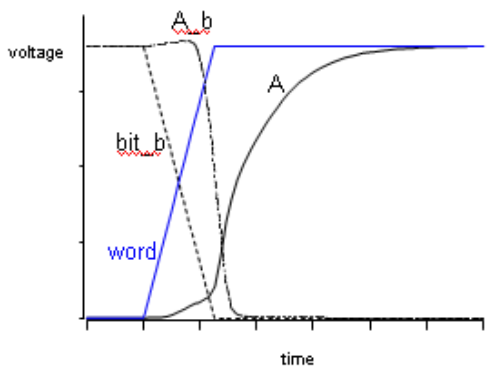


Figure 1.5: Write operation [8]

For a write operation, the cross couple inverters might need to be overpowered by the bitlines in order to switch state. P1 and P2 are easily overpowered since they are smaller. N1 and N3, however, are more difficult to overpower since they are most likely sized larger for read performance. Write performance is inversely proportional to the device sizes in the cross coupled inverters and their ability to maintain a logic state.

Stability comes into consideration when considering various operating conditions. In low power mode, the supply voltage to an SRAM cell might be lowered to reduce leakage current. If the cross-coupled inverters have been sized to maximize write performance, the memory cell might flip states in the presence of noise or in situations where 'word' is asserted but neither a read nor write is desired. Since the intent of this paper is to discuss differences between fully digital and sense amplifier read topologies, the 6T memory cell itself will be identical for all comparisons. Based approximations from Figure 1.2 and 1.3, the sizes used for analysis will be P1 = P2 = 0.2 µm, N2 = N4 = 0.2 µm, and N1 = N3 = 0.6 µm

Chapter 2

SRAM Array Construction

2.1 Bitline Height

The bitline height of an SRAM array is defined as the number of memory cells electrically connected to the 'bit' and 'bit_b' (Figure 1.1) nodes. This number of memory cells is influenced by cell leakage, density, desired performance, noise margin, and read/write circuitry. As the number of cells per bitline increases, the capacitance of 'bit' and 'bit_b' also increases. The increased bitline capacitance requires more time to discharge and results in slower read performance when using the same memory cell. Increasing the memory cell device sizes can increase performance but does so at the cost of density.

Figure 2.1 shows the leakage characteristics for a bitline with a column height of 64. Shown are bitline voltages both before and after precharge. No 'word' signal is asserted and therefore bitline droop is due purely to leakage through the memory cells' passgate device. The three curves represent memory cells using the three families of devices in IBM's 45nm multiple threshold device offering: high V_t (HVT), regular V_t (RVT), and super high V_t (SVT).

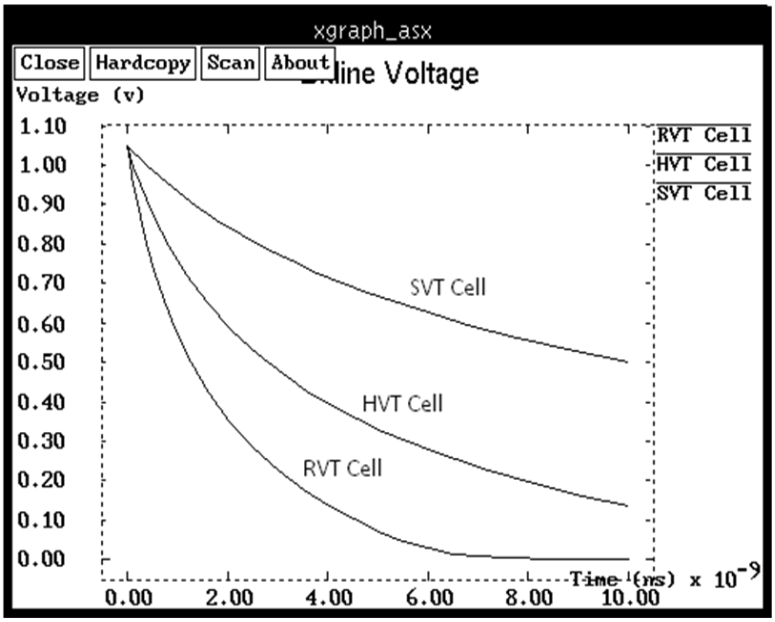


Figure 2.1: bitline droop due to leakage (cells identically programmed)

Figure 2.1 represents the worst case scenario where each memory cell stores identical states. Since this is not typical in memory arrays, Figure 2.2 shows bitline droop voltage characteristics where half of the memory cells on the bitline store a logic '0' and half store a logic '1.' The bitline droop due to leakage is less but still significant. Therefore, a keeper device will be needed to maintain bitline voltages. To further minimize non-active cell leakage effects, super-high V_t devices will be used and the cell will be approximately 0.6 µm tall and 1.2 µm wide.

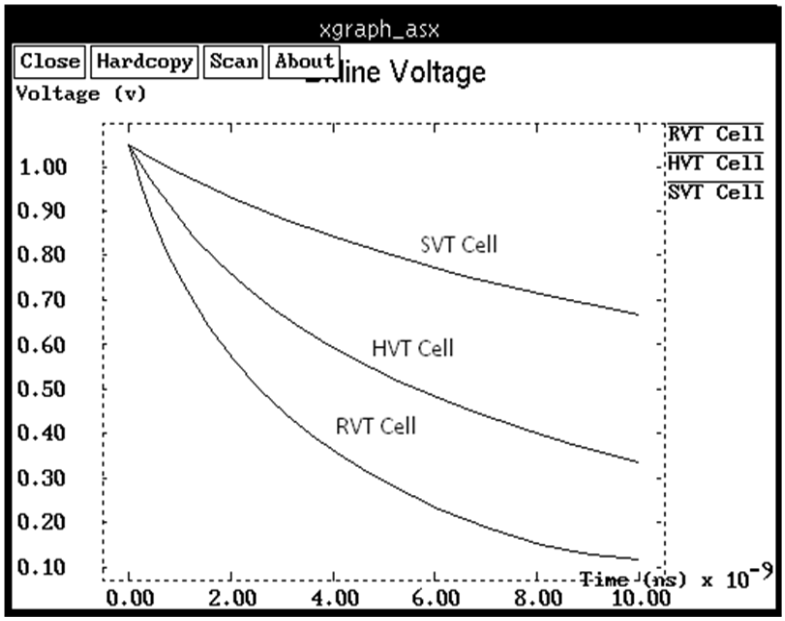


Figure 2.2: Bitline droop due to leakage (cells randomly programmed)

Figure 2.3 shows discharge trends for bitlines with different numbers of memory cells. With more memory cells per bitline, capacitive load increases and therefore slows down the discharge rate. In the 32 cells/bitline case, a partial swing SRAM array with a 50% voltage swing read requirement can start the read process 51 ps after the memory cell starts to discharge the bitline. In contrast, a fully digital SRAM array might require as much 90% swing before a logic gate can resolve the bitline. In the same 32 cells/bitline case, the read would start 100 ps after the bitline started to discharge. Table 2.1 shows a more complete delay Table for bitline discharging. '10% swing' is when the memory cell has discharged the bitline past Vss – (0.10)·Vss volts.

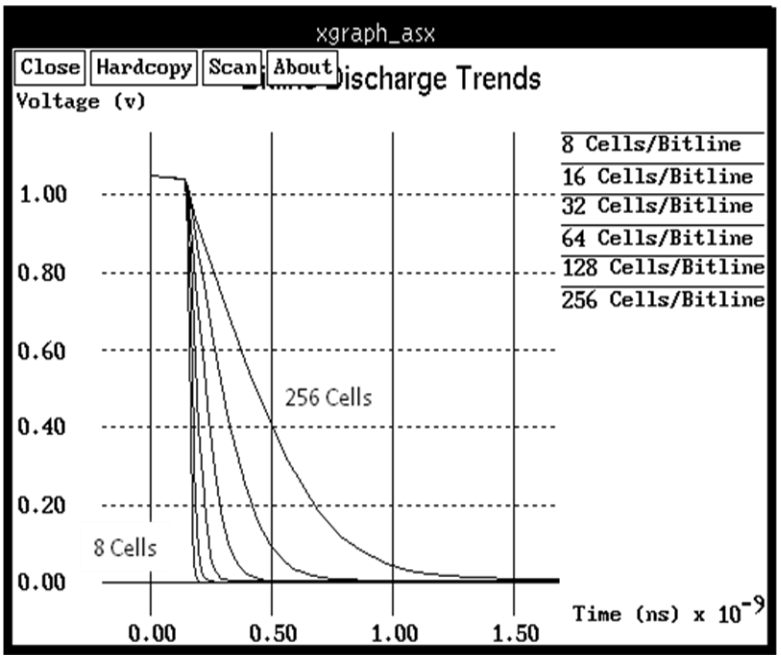


Figure 2.3: Discharge trends for multiple bitline heights

cells/ bitline	10% swing	50% swing	90% swing
8	17 ps	25 ps	38 ps
16	24 ps	34 ps	58 ps
32	31 ps	51 ps	100 ps
64	43 ps	85 ps	182 ps
128	66 ps	152 ps	349 ps
256	103 ps	276 ps	678 ps

Table 2.1: Discharge delay values for multiple bitline heights

2.2 Write Circuitry

The memory cell, though an integral part of the SRAM array's characteristics, is heavily influenced by the supporting circuitry. Write components must sample data from

upstream logic, synchronize that data to the timing of the array, and force that data onto the target memory cell with enough electrical strength such that the cell flips state within the allotted write time. If designed correctly, an SRAM array will hit and therefore read most of the time [11]. For this reason, the write circuitry should add as little parasitic capacitance to the bitlines as possible.

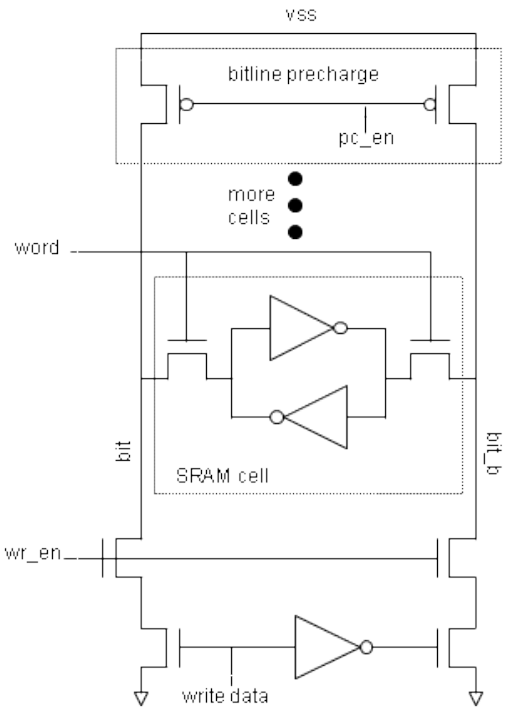


Figure 2.4: SRAM Write Circuitry

Figure 2.4 illustrates one method to write to an SRAM memory cell. The most notable feature is the double stacked NFET driving ‘bit’ and ‘bit_b.’ A static gate such as in inverter cannot be used to drive write data to the bitlines since they would interfere with read operations. For this reason, the write data driver must have a high impedance state. This can be achieved by driving ‘wr_en,’ in Figure 2.4, to logic '0.' Only one bitline is pulled low during a write, however. This topology assumes the capacitance of the non-driven bitline will remain at the precharge voltage until the write operation completes and the SRAM cell switches states.

During the time a bitline is not driven, voltage drift due to leakage (Figure 2.1/2.2) and noise effects can have catastrophic effects on write operations. If device leakage should create a weak current path to ground in the non-driven bitline, there might be insufficient voltage difference between ‘bit’ and ‘bit_b’ to change the state of the memory cell. Figure 2.5 remedies this by adding two cross-coupled PFET's that will keep the non-driven bitline at the voltage supply as soon one bitline is driven below a PFET threshold.

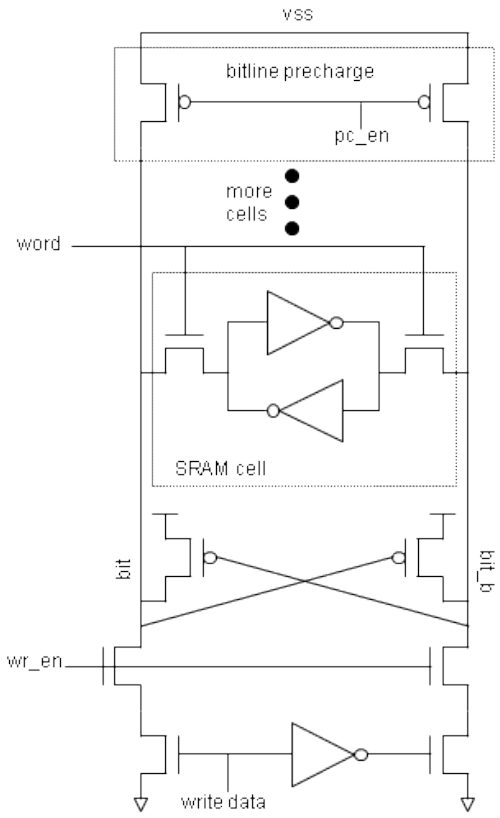


Figure 2.5: SRAM write circuitry with leakage protection

As mentioned earlier, the write circuitry for an SRAM array should add as little parasitic capacitance to the bitlines as possible. Figure 2.4 shows a configuration that reduces bitline capacitance due to the NFET pulldown devices. To achieve equal

strength, the pulldown NFET's in Figure 2.6 are only be half the size of the pulldown NFET's in Figure 2.5.

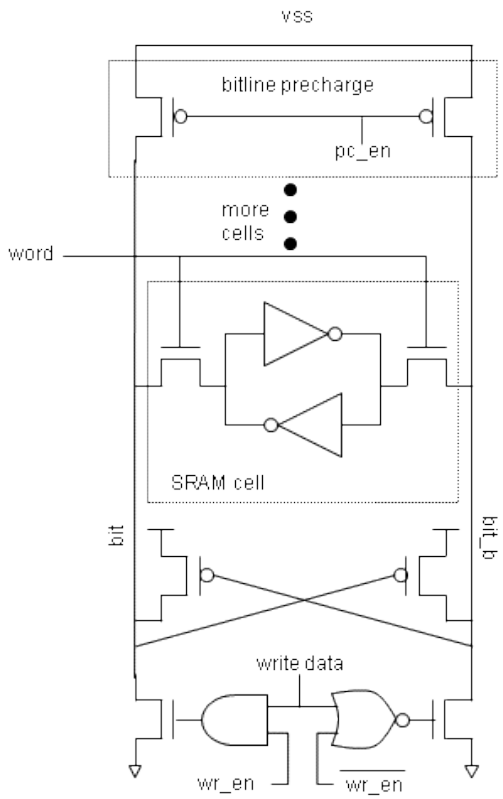


Figure 2.6: SRAM write circuitry reduced bitline capacitance

For fair comparison, the same write topology will be used for the full and partial swing designs. However, since the bitline capacitances for both will not be equal, the devices in the write circuitry will be sized to achieve similar write performance.

2.3 Full Swing SRAM Design

Figure 2.7 shows an example of a hierarchical, full swing read topology. The notion of a local bitline is introduced and, in this case, has three memory cells attached. Since there cannot be any electrical connection between the local bitlines, each must have

its own precharge and write circuitry. Implementing complementary global bitlines is optional and can be used for to increase noise immunity or performance. For purposes of this paper, only a single global bitline will be implemented.

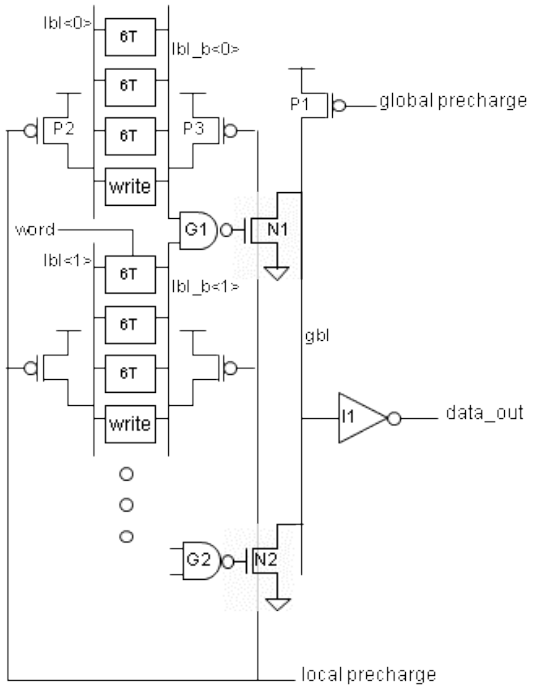


Figure 2.7: Full swing SRAM organization

NAND gates N1 and N2 from Figure 2.7 are sized for performance and noise rejection. They are meant to react to a '1' -> '0' transition since the local bitlines are precharged to the voltage rail. Through beta ratio adjustment, the threshold of the NAND gates can be raised to increase performance or lowered to increase noise immunity.

Figure 2.8 is a timing diagram of a full swing read operation. The global and local precharge signals start off at a logic '0' and drive both global and local bitlines to the voltage rail. To avoid collision, the local precharge devices (*e.g.*, P2, P3) must turn off before a read occurs. If the memory cell tries to discharge a local bitline during precharge, a collision would result and the global NAND gate (*e.g.*, G1, G2) might not be

able to evaluate the local bitline correctly. After local precharge is disabled (‘local precharge’ = '1'), a read starts with the assertion the signal 'word.' Only a single row will have its wordline signal activated. For this reason, any local bitline discharge on a complement local bitline will result in a global bitline discharge. Due to circuit delays, the global precharge can be disabled after the local precharge. In addition, the global precharge should be enabled after local precharge to hold valid information at ‘data_out’ for as long as possible.

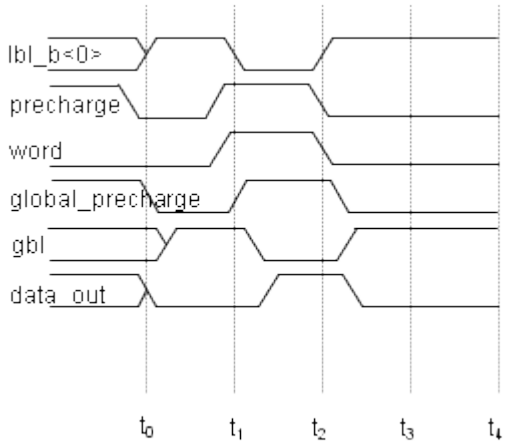


Figure 2.8: Timing diagram, full swing SRAM read

2.4 Partial Swing SRAM Design

Figure 2.9 shows an example of an SRAM memory array column that employs partial swing read. More memory cells are dotted onto a bitline and therefore each memory cell must discharge more capacitance during a read. Since the memory cell's ability to sink current is constant, the bitline will discharge more slowly than in the full swing case. Due to cycle time constraints, the active bitline (‘bl_b’ in the case of Figure 2.10) will not discharge completely. This is not a problem since the sense amplifier does not require full swing at its inputs to evaluate read data. As also shown in Figure 2.10, the precharge must turn off complete before read can begin in order to avoid collision. In the partial swing read, collision avoidance is more important due to the limited bitline swing.

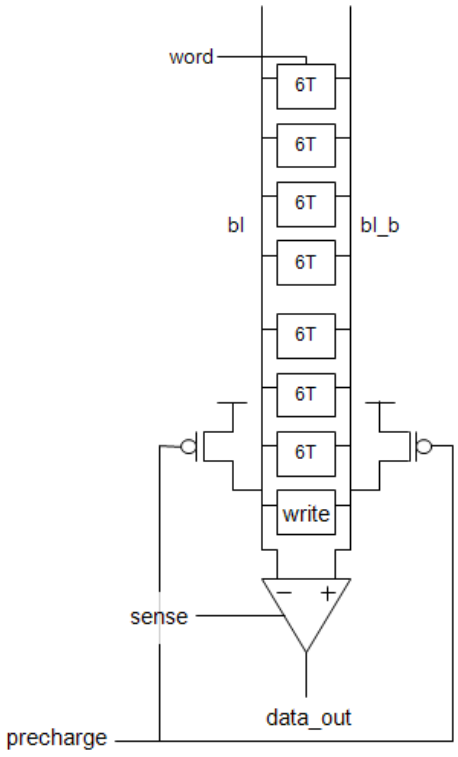


Figure 2.9: Partial swing SRAM organization

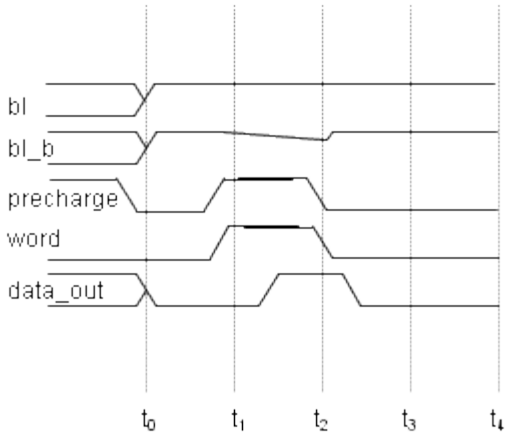


Figure 2.10: Timing diagram, partial swing SRAM read

2.4.1 Sense Amplifier Design

The sense amplifier in partial swing SRAM memories can be implemented in multiple ways. Perhaps the most straightforward is shown in Figure 2.11. This structure utilizes a current-steered, cross-coupled inverter pair to amplify any difference in voltage between ‘bl’ and ‘bl_b’ [4]. However, this structure works regardless of operation and will even amplify noise. This can result in wrong data being sampled downstream and would be particularly harmful if that sampling mechanism was dynamic.

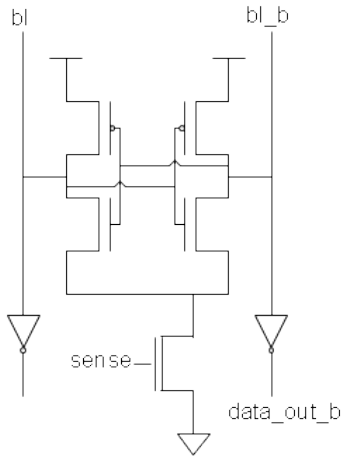


Figure 2.11: Typical sense amplifier design [5]

With the sense amplifier from Figure 2.11, there is no separation between the cross-coupled inverters and the bitlines. Therefore, the cross-coupled inverters would also need to drive the bitlines to opposite voltages once the switch threshold is reached. This would slow down the propagation of bitline values to ‘data_out’ or ‘data_out_b.’

For this reason, isolation circuitry can be added as shown in Figure 2.12. With this implementation, the sense amplifier is not enabled until ‘bl’ and ‘bl_b’ have developed sufficient difference in voltage. ‘Sense’ would be asserted some time after the wordline pulse. This amount of time depends on the memory cell's ability to discharge

the bitline and the sense amplifier's differential threshold. 'Sense' should only be asserted when the voltage difference between 'bl' and 'bl_b' is such that a read is known to be happening. This time is determined through circuit simulation of various operating conditions or even determined post-hardware with the use of an adjustable delay for 'sense.'

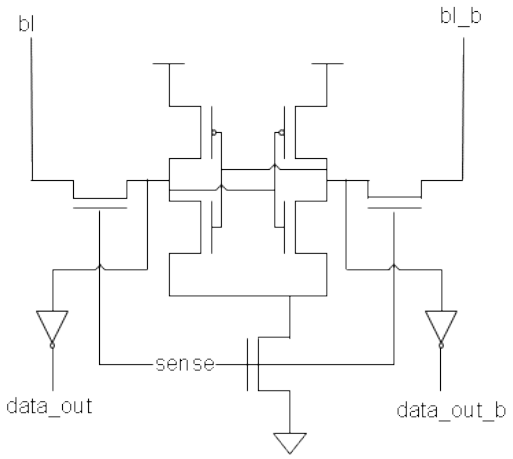


Figure 2.12: Sense amplifier design with isolation

The isolation of bitlines from cross-coupled inverters can be improved from Figure 2.12. In this sense amplifier implementation, each bitline must drive four diffusion capacitances and four gate capacitances – the majority of which is on the other side of an isolation passgate. In partial swing configurations, bitline loading should be as little as possible. Figure 2.13 shows a sense amplifier configuration that only adds one gate capacitance load per bitline. It has the same current steering mechanism to change the state of the cross-coupled inverters and therefore has similar threshold characteristics.

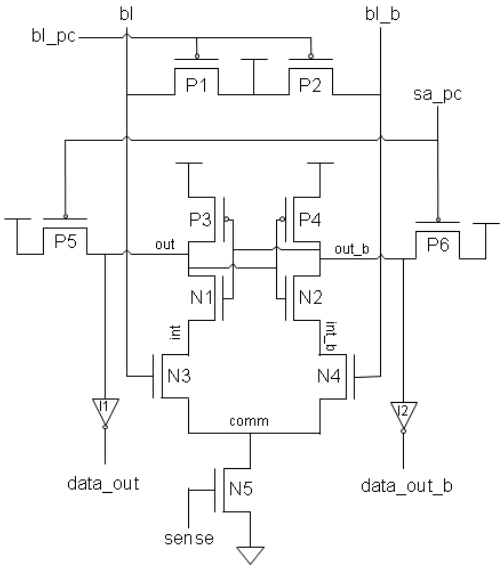


Figure 2.13: Sense amplifier design with improved bitline isolation [9]

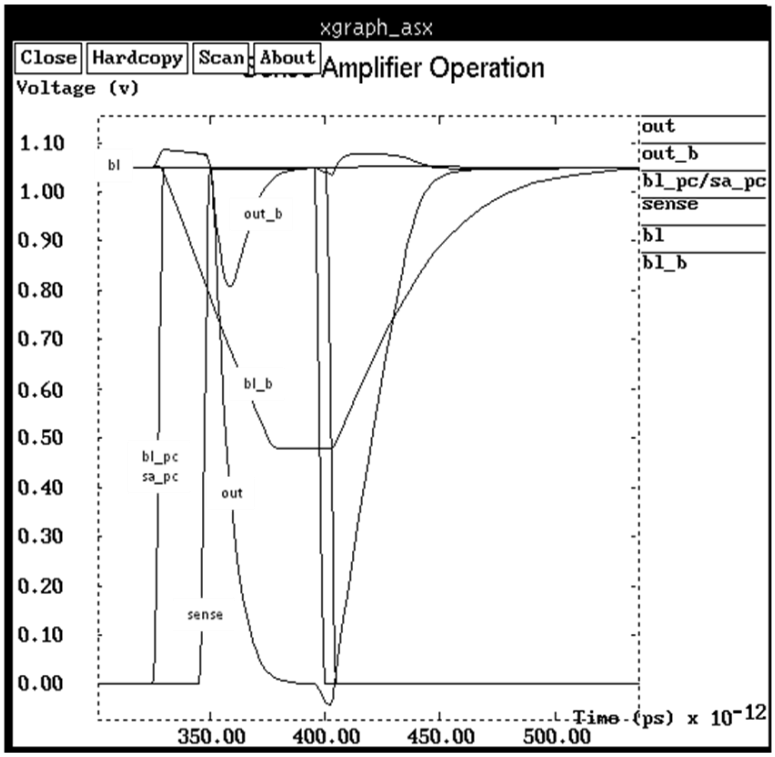


Figure 2.14: Sense amplifier design with improved bitline isolation operation

One notable detail in Figure 2.13 is the precharge device connected to the internal nodes of the cross-coupled inverters. This is required to reset the state held from the previous read. To prepare the sense amplifier for a read operation, 'sa_pc,' is therefore driven to logic '0.' The bitlines must also be precharged and therefore 'bl_pc' is also driven to logic '0' before a read. Lastly, 'sense' must be driven to a logic '0' such that no nodes in the sense amplifier discharge before intended. Before the read takes place, N1, N2, N3, and N4 are in conductive state and nodes 'int,' 'int_b,' and 'comm' are pulled within a V_t of the voltage rail by P5 and P6. Both 'bl_pc' and 'sa_pc' are then de-asserted.

A bitline discharge will start to turn off either N3 or N4. 'Sense' is asserted sometime after one bitline starts to discharge and turns on N5 completely. If 'bl_b' discharges, for example, N4 will conduct less current than N3 when N5 turns on. When enough voltage difference between 'bl' and bl_b' has developed and 'sense' is asserted, both 'out' and 'out_b' will start to discharge. As N3 conducts more current than N4, however, 'out' will discharge faster than 'out_b.' As 'out' discharges, P4 starts to turn on and pulls 'out_b' back to the voltage rail as shown in Figure 2.14. This causes N1 to turn on more aggressively since its gate was previously driven at one V_t below rail voltage. This accelerates the rate at which 'out' is discharged, turns on P4 more strongly, and aggressively pulls 'out_b' to voltage rail. N1 is now completely turned on, 'out' is fully discharged, and the sense amplifier has locked into a finite logic state.

The sense amplifier in Figure 2.13 is incredibly sensitive to voltage differences between 'bl' and 'bl_b' and cannot differentiate between transient noise and bitline discharge due to read activity. Simulations show that differences as little as 10 mv cause the cross-coupled inverters to lock into a state. In addition, once the cross-coupled inverters have locked, only the next cycle's precharge can clear the state. Even if the bitlines recover from noise and correct the errant voltage differential, the sense amplifier would still output the incorrect value.

As stated earlier, ‘sense’ cannot be asserted immediately after precharge. Analysis must be done to understand maximum possible bitline noise. In addition, the rate at which the memory cell discharges the bitline must also be known. Time must be allowed for one bitline to discharge to a level where the voltage difference between the bitlines is known to be due to memory cell activity and not noise. If there is possibility for large bitline noise, ‘sense’ must be delayed significantly to allow a greater bitline voltage differential to develop. However, performance would be lost as downstream logic will wait longer to have valid data from the read operation.

Internally, the glitch on the non-switching node of the sense amplifier must be reduced as possible to avoid errors. By schmoo’ing ‘sense’ delay, a point of diminishing returns (less glitch is better) can be seen from the plot in Figure 2.15.

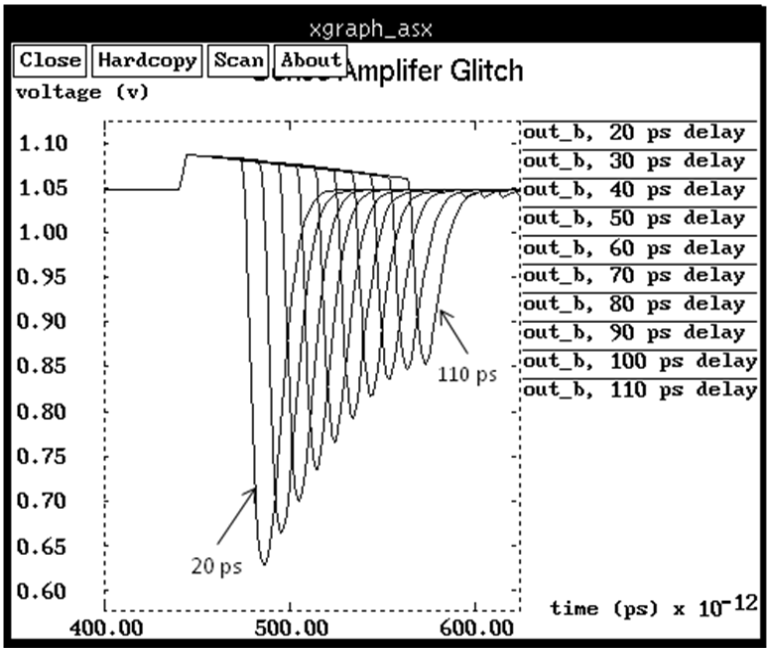


Figure 2.15: Sense amplifier glitch with various ‘sense’ delay settings

At 80 ps, increasing the delay in ‘sense’ does not decrease the glitch voltage as effectively. Therefore, the amount of delay for ‘sense’ will be set at 80 ps. The glitch

data was collected with the actual 128 tall bit column and therefore used the correct bitline discharge rates.

2.5 Address Predecode and Decode

The bitline structure in an SRAM array is supported by many other circuits. There are wordline drivers that must activate the memory cells, precharge drivers that clear bitlines and prior read data, sense enable drivers that turn on the sense amplifier in the partial swing case, and decode circuitry that translates the address into a unique row number. These pieces of circuitry also affect performance and must remain as tranparent as possible in comparing partial swing and full swing SRAM designs.

A 128 x 128 SRAM array will be used for analysis. This size will exaggerate capacitive loading and slow down edge rates so that differences will be easy to observe. 128 entries will require seven bits of addressing. Simulations will be set up assuming the addressing scheme described in Figure 2.16.

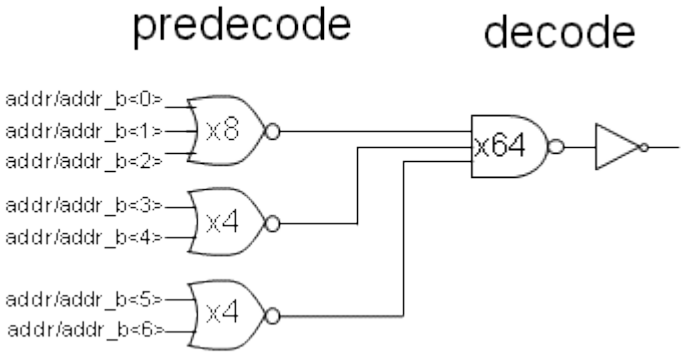


Figure 2.16: Predecode/decode scheme [10]

With a short cycle time, the predecode output might have trouble reaching full rail. This is due to expected high output loading and necessitates address circuitry partitioning as shown in Figure 2.17.

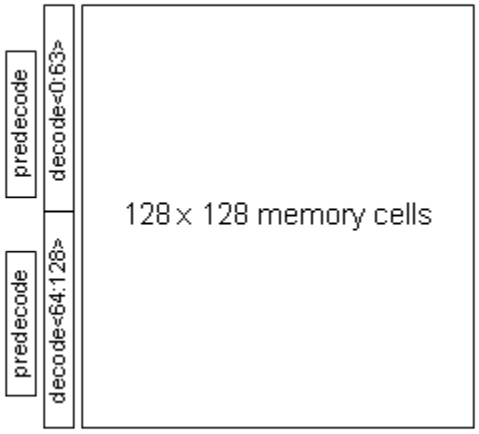


Figure 2.17: Address predecode/decode placement

The predecode/decode will be implemented with 16 NOR gates, 64 NAND gates, 1 drive inverter, and will be instantiated twice to limit predecode output loading. Assuming 1.5x (1.5 x minimum width) metal three wiring for the 'word' signal and 5x metal two for predecode wiring, the address decode timing is described by Figure 2.18. 'addr_in' to 'predec_out' delay is 87 ps and 'predec_out' to 'word' delay is 56 ps for a total of 144 ps of address decode delay.

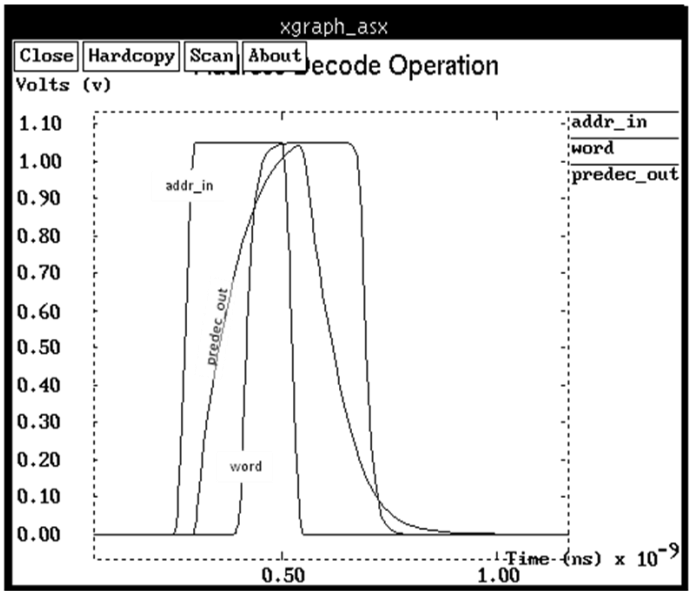


Figure 2.18: Predecode/decode operation

Chapter 3

Full Swing/Partial Swing SRAM Comparison

3.1 Performance Comparison

To study the read performance of the full swing SRAM, five different scenarios will be analyzed. An array with 4 cells, 8 cells, 16 cells, 32 cells, and 64 cells per local bitline will be evaluated and the best performance of the five will be used to compare against the partial swing SRAM read. The local bitline discharge behavior of the full swing SRAM is similar to that shown in Figure 2.3. With more cells per bitline, the discharge rate slows down since the bitline capacitance increases but the cell's ability to discharge a bitline does not. The performance of the partial swing SRAM will be easier to evaluate since all 128 memory cells will be dotted together on the same bitline. 'Sense' will be delayed as determined through the internal glitch minimization technique. With both cases, all precharge will be disabled before 'word' is enabled so that collision will not affect performance. The delay measurement starts when the address input is valid and ends when valid data is available.

3.1.1 Partial Swing SRAM Read Performance

The read operation of the partial swing SRAM array is shown in Figure 3.1. The bitline continues to discharge after the 'sense' has been de-asserted to provide 50% voltage crossing and delay values. The delay values are shown in Table 3.1 show that data is read from the bitline and is available in buffered form 247 ps after address is valid. The individual delay components are described in Table 3.1.

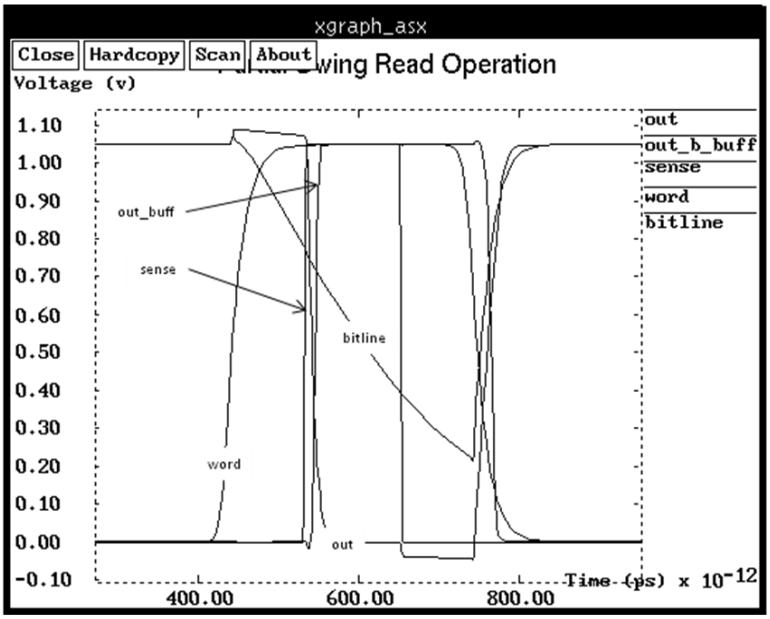


Figure 3.1: Partial Swing Read Operation

component	50%-50% delay
in	0 ps
predec_out	87 ps
word	144 ps
sense	234 ps
bitline	300 ps
out	243 ps
out_b_buff	247 ps

Table 3.1: Partial Swing SRAM Delay table

3.1.2 Full Swing SRAM Read Performance

The local bitline discharge behavior in the full swing SRAM is described in Figure 3.2. As expected, with more cells per bitline, the discharge happens more slowly.

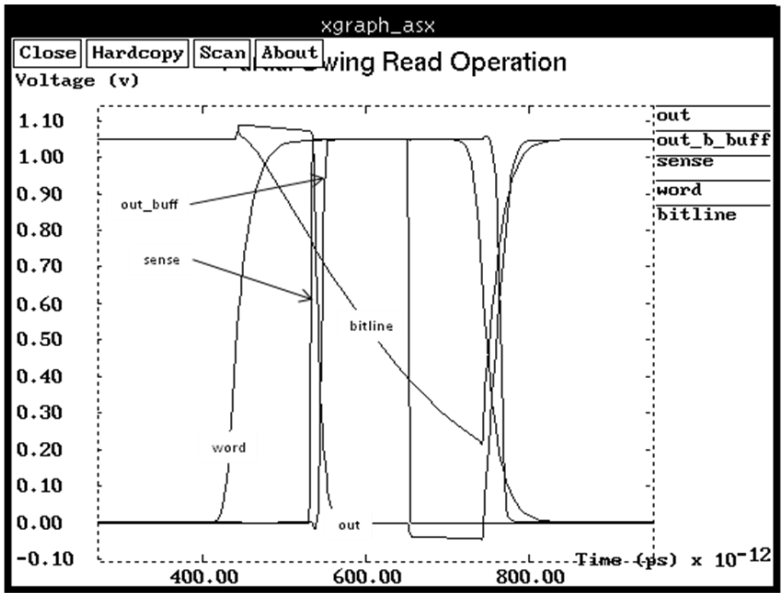


Figure 3.2: Local bitline behavior, full swing SRAM

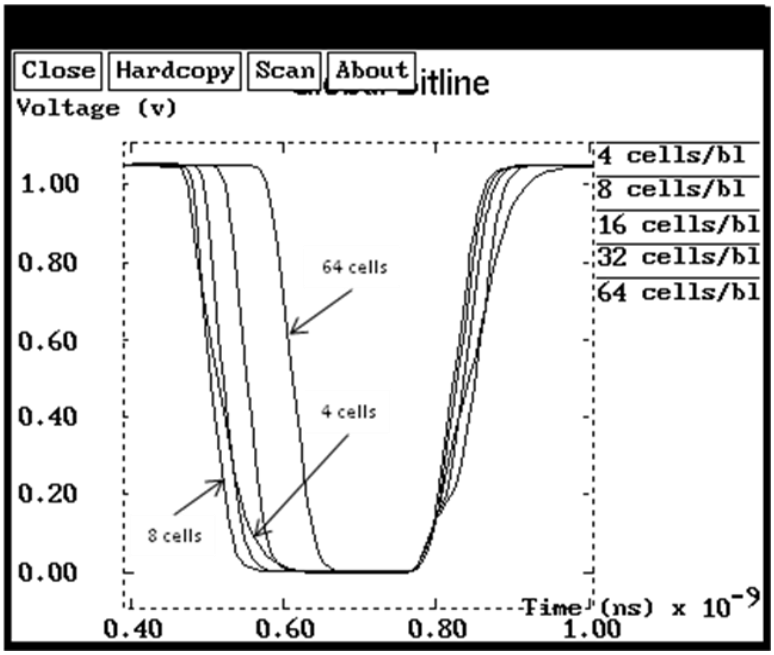


Figure 3.3: Global bitline behavior, full swing SRAM

The global bitlines in the full swing case do not follow the same trend as the local bitlines. The green line in Figure 3.3 represents four cells per bitline where the magenta lines represents eight. The falling edge represents the bitline discharge due to memory cell evaluation and will be used to gauge performance. The four cell case is actually slower than the eight cell case due to global bitline loading. With more cells per bitline, there are less global bitline pulldown NFET's (N1 and N2 from Figure 2.7). With less diffusion capacitance, the global bitline discharges more quickly. This effect disappears with sixteen cell case as the local bitline slowdown is greater than the global bitline speedup. Figure 3.4 illustrates the effect to output timing with the eight cells per case outperforming the four cells per bitlines case.

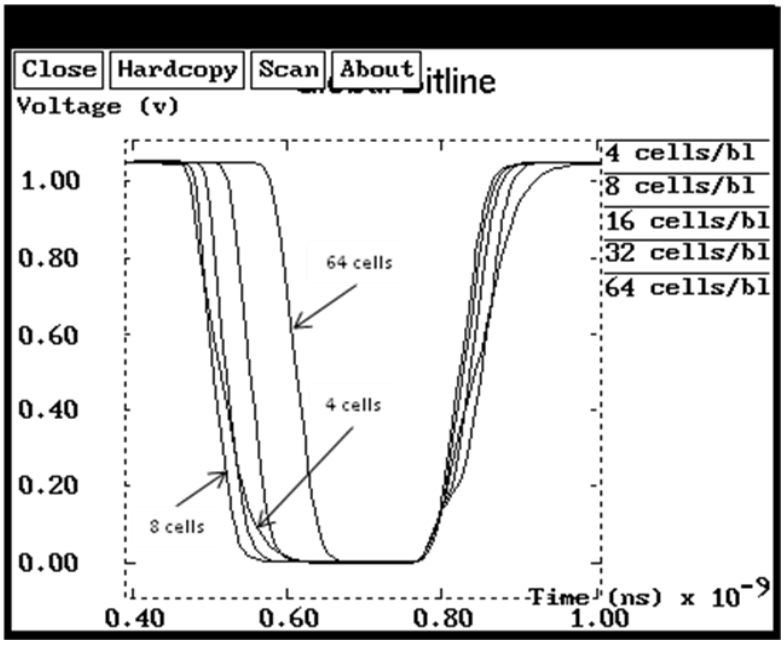


Figure 3.4: Output delays, full swing SRAM

The operation and timing information for the full swing SRAM is described in Figure 3.5 and Table 3.2. In comparing performance between the full swing and partial swing SRAM reads, there is no clear advantage in either method. There is an 11 ps advantage in the full swing case but this can easily be eliminated be enabling the sense

amplifier from the partial swing case sooner. This would decrease noise immunity and might not be feasible for certain applications. In later analysis, noise immunity of both SRAM read methods will be compared to determine if such tuning is possible.

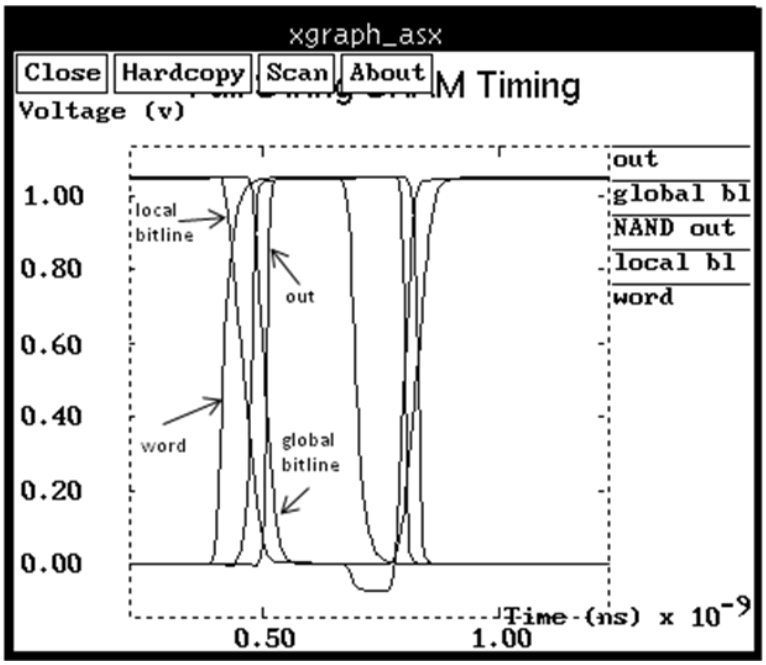


Figure 3.5: Full swing SRAM read operation

component	50%-50% delay
in	0 ps
predec_out	87 ps
word	144 ps
local bitline	191 ps
NAND out	203 ps
global bitline	230 ps
out_buff	236 ps

Table 3.2: Full Swing SRAM delay table

3.2 Power Comparison

To measure switching power, current was monitored while the circuit was stimulated with a pattern that forced switching activity in every cycle. The PowerSPICE code used to derive energy and power can be found in Appendix B.1. The models used for simulation contained provisions for leakage power that inflated switching power values and had to be removed. Leakage power was compared by looking at resultant leaking device widths. The leaking device width takes into account raised intermediate voltages due to stacked devices, state dependency, and input state probability.

3.2.1 Switching Power

To avoid state-dependent leakage power from polluting the switching power value, power was measured at two different frequencies. The final switching power was then derived with Formulas 1, 2, 3, and 4.

$$p = \tfrac{1}{2}cv^2f \qquad \text{Formula 3.1}$$

$$p_{total} = \tfrac{1}{2}cv^2f + p_{leak} \qquad \text{Formula 3.2}$$

$$p_1 = \tfrac{1}{2}cv^2f_1 + p_{leak},\ p_2 = \tfrac{1}{2}cv^2f_2 + p_{leak} \qquad \text{Formula 3.3}$$

$$c = 2 \cdot \frac{P_1 - P_2}{v^2 f_1 - v^2 f_2} \qquad \text{Formula 3.4}$$

Leakage and capacitance should be constant across all valid frequencies of operation. As shown in Formula 3.4, effective capacitance can be extracted without a leakage component. Applying Formula 3.1 to the capacitance calculated from Formula 3.4 results in the true switching power.

Full Swing SRAM AC Power Rollup	measured power @0.67 ghz (w)	measured power @1.33 ghz (w)	Ceff (f)	instances	switch prob.	switching power @1.5 ghz (w)	switching power @2.0 ghz (w)	switching power @2.5 ghz (w)
global bitline precharge/keeper	2.32E-05	4.46E-05	5.81E-14	128	0.5	3.07E-03	4.10E-03	5.12E-03
global bitline precharge driver	1.82E-04	3.47E-04	4.50E-13	1	1.0	3.72E-04	4.96E-04	6.20E-04
input address driver	3.29E-04	6.45E-04	8.59E-13	7	0.5	2.49E-03	3.31E-03	4.14E-03
local bitline joining NAND	1.07E-05	1.78E-05	1.93E-14	1024	0.1	1.02E-03	1.36E-03	1.70E-03
local precharge	8.18E-06	1.61E-05	2.16E-14	4096	0.3	1.83E-02	2.44E-02	3.05E-02
local precharge driver	1.87E-04	3.57E-04	4.63E-13	16	1.0	6.13E-03	8.17E-03	1.02E-02
address decode NAND/driver	3.73E-04	7.21E-04	9.47E-13	128	0.0	7.83E-04	1.04E-03	1.31E-03
address predecode NOR	1.25E-03	2.40E-03	3.14E-12	3	1.0	7.79E-03	1.04E-02	1.30E-02
cell power	3.95E-05	3.97E-05	6.51E-16	16384	0.0	6.89E-05	9.19E-05	1.15E-04
					total	4.00E-02	5.34E-02	6.67E-02

Table 3.3: Full Swing SRAM switching power rollup

Partial Swing AC Power Rollup	measured power @0.67 ghz (w)	measured power @1.33 ghz (w)	Ceff (f)	instances	switch prob.	switching power @1.5 ghz (w)	switching power @2.0 ghz (w)	switching power @2.5 ghz (w)
input address driver	3.24E-04	6.35E-04	8.48E-13	7	0.5	2.45E-03	3.27E-03	4.09E-03
address predecode NOR	1.23E-03	2.37E-03	3.08E-12	3	1.0	7.65E-03	1.02E-02	1.28E-02
sense driver	3.10E-04	5.89E-04	7.57E-13	1	1.0	6.26E-04	8.35E-04	1.04E-03
bitline precharge driver	3.76E-04	7.36E-04	9.79E-13	1	1.0	8.09E-04	1.08E-03	1.35E-03
address decode NAND/driver	3.68E-04	7.10E-04	9.31E-13	128	0.0	7.70E-04	1.03E-03	1.28E-03
cell power	5.11E-05	5.11E-05	1.45E-16	16384	0.0	1.54E-05	2.05E-05	2.56E-05
sense amplifier/bitline precharge	5.77E-05	1.10E-04	1.44E-13	128	1.0	1.52E-02	2.03E-02	2.53E-02
					total	2.75E-02	3.67E-02	4.59E-02

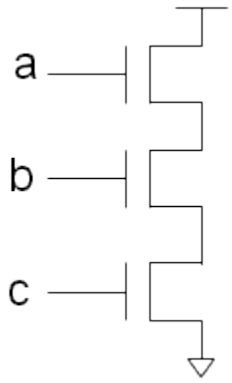
Table 3.4: Partial Swing SRAM Switching power rollup

Table 3.3 and 3.4 rollup the switching power components of both SRAM read topologies for different frequencies of operation. The two measured columns are the two data points needed to extract C_{eff} from measured power values without leakage power. The 'instances' column represents the number of instantiations of each element needed to assemble a 128 x 128 array. The 'switch probability' column is the likelihood the

described instance will switch during a read operation. For example, a 128 x 128 array will contain 16384 memory cells. During any read, only one row will be active. In that row, each cell will discharge the true or complement bitline. Therefore the product of the switch probability and number of instances should equal one hundred twenty eight. The remaining probabilities are calculated assuming a 50% distribution of logic '1's and '0's stored in the memory cells.

3.2.2 Leakage Power

Absolute leakage power values were not publicly available for IBM's 45 nm process. Therefore, relative comparisons based on total leakage width were used instead. Leaking device widths are calculated assuming all precharge inputs are logic '0' and other inputs are random. Super-high V_t devices are de-rated by a factor of ten in comparison to regular V_t devices to reflect reduced leakage per unit length. Device stacking factors are factored into leakage width as shown in Figure 3.5. N1, N2, and N3 are identical in dimension [13].



a	b	c	resultant leakage width
0	0	0	neglible
0	0	1	10% of N2
0	1	0	10% of N3
0	1	1	100 % N1
1	0	0	10% ofN3
1	0	1	80% of N2
1	1	0	80% N3
1	1	1	n/a

Figure 3.5: Stacked device leakage table

Resultant leaking width in stacked devices can be reduced due to leakage current's dependence on source to drain voltage. As source-drain voltage across a disabled device decreases, the leakage current decreases as well. From simulation,

intermediate voltages in stacked devices can be determined and scaling factors can be calculated. There are two main effects that influence the scale factor. The fact that an NFET's and PFETs's cannot pass the opposite rail voltage without a V_t drop and that disabled devices, when stacked, pass very little leakage current.

Using weighted averages to take into account random inputs and the scaling factors shown in Figure 3.5, the resultant leakage widths for the partial swing and full swing SRAM components are listed in Table 3.5 and 3.6.

Full Swing SRAM DC Rollup	leaking NFET width (um)	leaking PFET width (um)	instances	leaking NFET width per array (um)	leaking PFET width per array (um)
global bitline precharge/keeper	0	0	128	0	0.0
global bitline precharge driver	4.2	26.2	1	4.2	26.2
input address driver	5	6.5	7	35	45.5
local bitline joining NAND	0	1.2	128	0	153.6
local precharge	0	0	4096	0	0.0
local precharge driver	4.2	26.2	16	67.2	419.2
address decode NAND/driver	40.29	20.34	128	5157.12	2603.5
address predecode NOR	31.5	81.09	6	189	486.5
cell	0.02	0.08	16384	327.68	1310.7
			total	5.78E+03	5.02E+03

Table 3.5: Full swing SRAM leakage width rollup

Partial Swing DC Rollup	leaking NFET width (um)	leaking PFET width (um)	instances	leaking NFET width per array (um)	leaking PFET width per array (um)
input address driver	5	6.5	7	35	45.5
address predecode NOR	31.5	81.09	6	189	486.5
sense driver	8.55	52.364	1	8.55	52.4
bitline precharge driver	4.216	26.2	1	4.216	26.2
address decode NAND/driver	40.29	20.34	128	5157.12	2603.5
cell	0.02	0.08	16384	327.68	1310.7
sense amplifier/bitline precharge	0	2.4	128	0	307.2
			total	5.72E+03	4.83E+03

Table 3.6: Partial swing SRAM leakage width rollup

3.3 Noise Rejection

Since noise can be unpredictable in terms of phase and magnitude, noise was injected into partial swing and full swing SRAM bitlines at 10 ps offset intervals and at magnitudes ranging from 10 mv to 1000 mv. The 10 ps intervals ensured that noise would be present during the worst possible portion of the read cycle. If noise was present during precharge, for example, there would be little consequence. If noise was present in the partial swing SRAM case when ‘sense’ is enabled, the consequences might be very different. The duration of the noise was also varied between 10 ps and 90 ps to modulate the amount of energy delivered by the noise pulse.

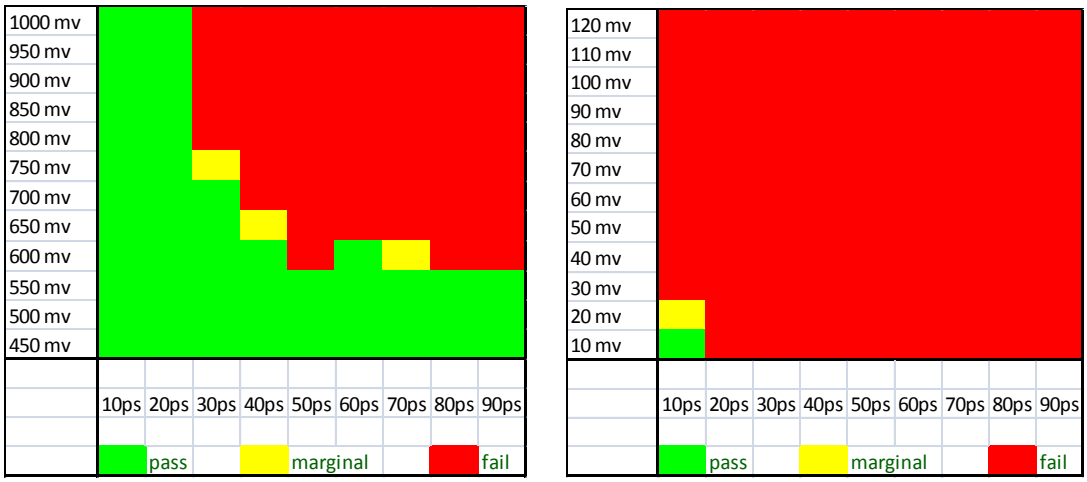


Figure 3.7: Pass/fail plot of full swing(L) and partial swing(R) SRAM's under noise

Figure 3.7 shows pass/fail data points for the two SRAM topologies in the presence of noise. A fail is defined as mismatch between output data in the presence of noise and output data without noise. A fail at any phase offset constitutes a fail for that particular noise duration and magnitude. It is clear that the full swing SRAM has much more noise rejection capability than the partial swing SRAM. The full swing SRAM pass/fail point seems to start at 600 – 650 mv. This is most likely dictated by the NAND (G1 and G2 from Figure 2.5) gate's threshold voltage. If the noise magnitude does not surpass the NAND gate's threshold, no false switch will result. The sensitivity of the sense amplifier is the primary culprit in the partial swing SRAM's lack of noise rejection capability. The sense amplifier, with appropriate 'sense' timing, was observed to switch with a voltage differential as little as 10 mv. Since noise was injected at multiple phases, it was likely that a noise pulse occurred at a time most likely to induce a fail.

Chapter 4

Conclusions

4.1 Additional Design Considerations

The data presented characteristics of partial swing SRAM's and full swing SRAM's. Power seemed to be the only area with a clear winner. With fewer bitlines to swing, less voltage to swing them, and less circuitry than its competition, the partial swing SRAM had an advantage on power that probably cannot be reversed by design improvements. With the remaining two comparison points, however, there seemed to be many tweaks that could be done to make either design the winner.

In regards to performance, the data showed the full swing design to be faster. However, devices with different V_t's could have been used to improve the response of the sense amplifier. A hierarchical bitline design where half of the memory cells are located above the sense amplifier and half are located below could have been implemented to reduce bitline wire delay and increase bitline discharge rate. Combined with more aggressive 'sense' timing, the partial swing design probably could have been designed to be faster than the full swing SRAM.

With respect to noise rejection, a different sense amplifier design with better common mode rejection could have been used. In combination with a twisted bitline design, noise sensitivity could be greatly reduced. In addition, integration tactics could be implemented that surround the bitlines with quiet metals or metals with known noise phase. This would reduce the possibility of noise events while the sense amplifier is enabled and even further decrease noise sensitivity of the partial swing design.

4.2 Choosing an SRAM Topology

Figure 4.1 combines each of the three comparison criteria into theoretical designs.

Each of the three comparison criteria is given a ‘priority’ or ‘non-priority’ classification. The groupings are then subjectively assigned a SRAM topology that would best fit the design constraints.

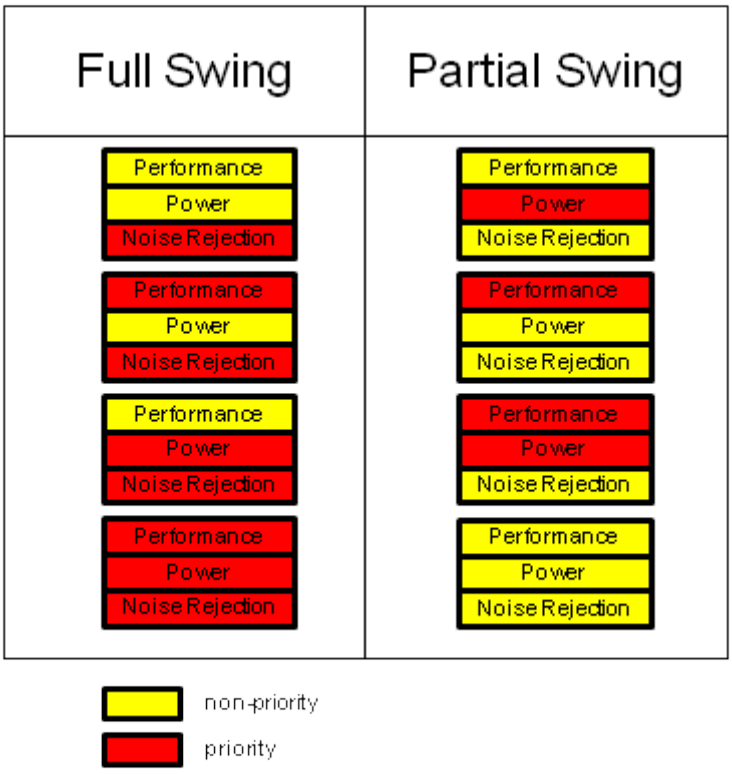


Figure 4.1: SRAM topology selection

The designs that have only one priority constraint are easy to separate into the full swing or the partial swing group. With power, for example, the data shows the partial swing SRAM consumes less and should be chosen if power savings is vital.

If performance is most important, the delay value for ‘sense’ can be reduced to significantly increase the partial swing SRAM performance. The amount of performance that can be gained by speeding up the ‘sense’ delay is greater than what can be gained by threshold tuning in the full swing case. When performance is the only priority, the partial

swing SRAM should also be chosen. Consequently, when power and performance are higher in priority, the partial swing SRAM is also chosen.

With the partial swing SRAM offering very little noise robustness, the designs where noise immunity is a priority are best served by the full swing SRAM design. Wiring and integration techniques can help but do not increase the actual noise rejection capabilities of the partial swing SRAM.

Choosing between partial swing and full swing is more difficult when two or more of the comparison criteria are high priority. The noise requirement should take precedence over power and performance. Noise modeling has been historically difficult to model and predict [12]. Since power and performance can be well characterized with accurate workloads, a full swing SRAM should be used when noise rejection is important.

In the end, the choice between full swing and partial swing SRAM designs must take into account more than power, performance, and noise robustness. Factors like cost, design complexity, area, and migratability also come into play. Unfortunately, the latter factors require some level of physical design to quantify. The goal of this paper is to provide insight into the decision making process when choosing between full swing and partial swing SRAM's without doing a complete design. With the data presented and the methods used for analysis, that choice should be possible.

Appendix A

Schematics

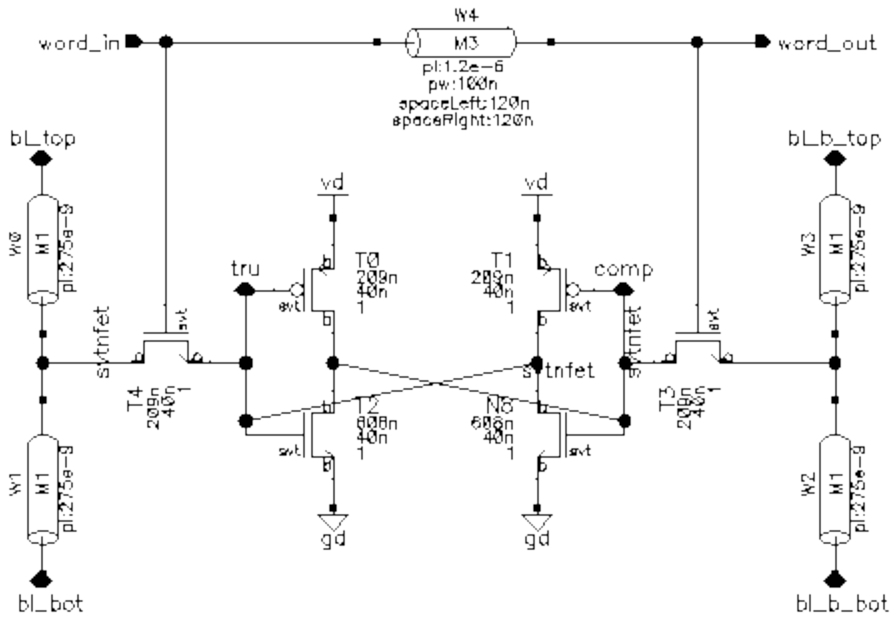


Figure A.1: SRAM cell with parasitic word and bitline loading

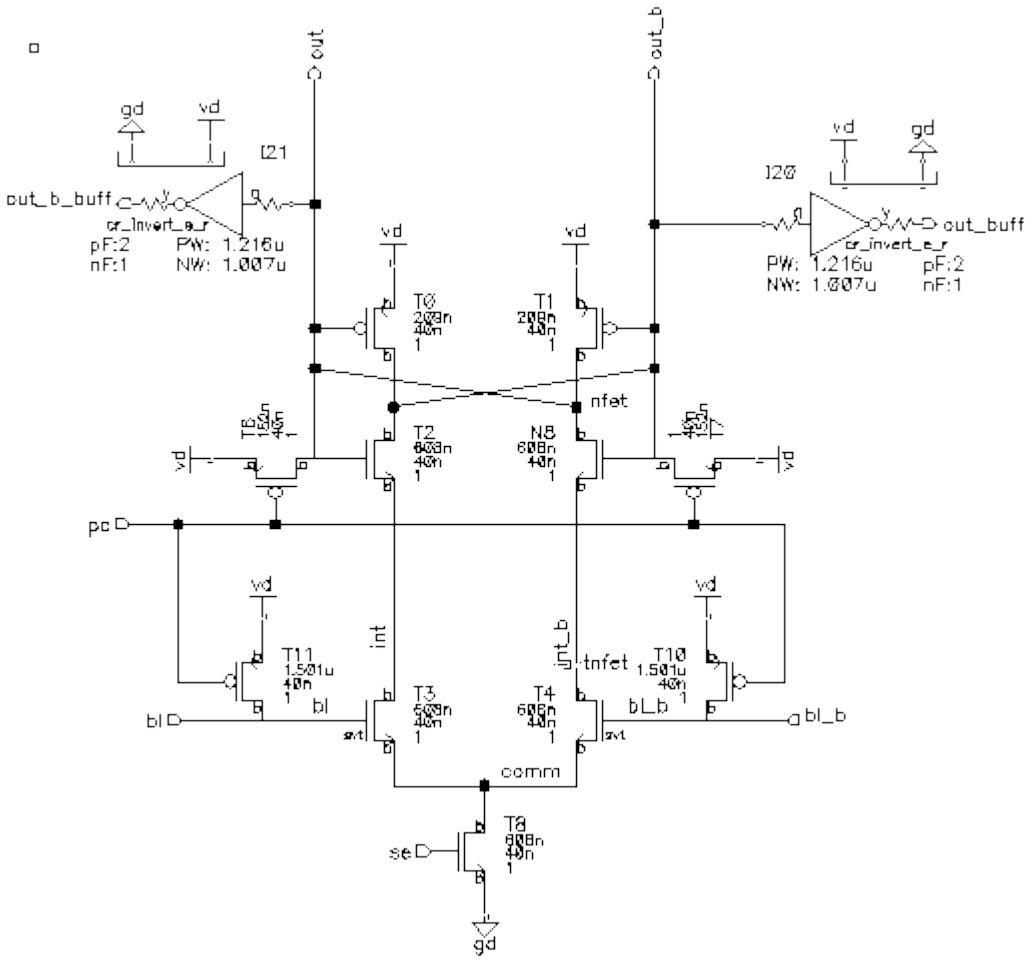


Figure A.2: Sense amplifier with bitline precharge

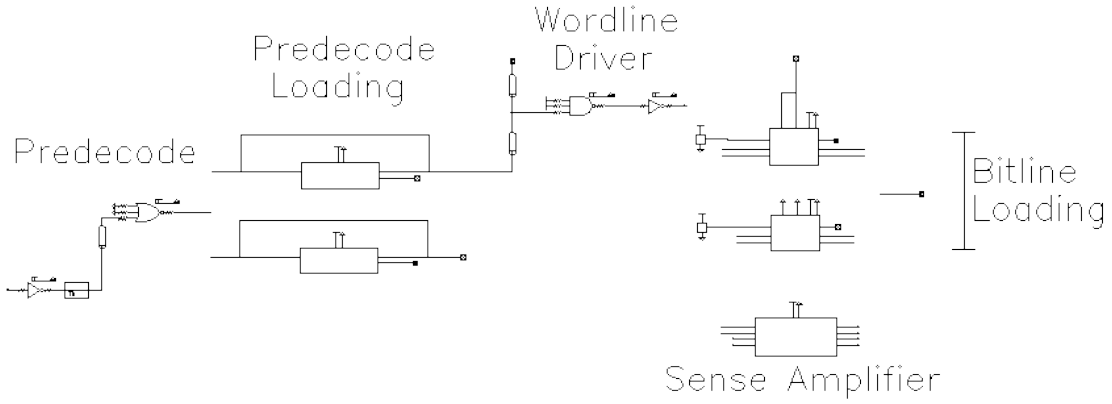
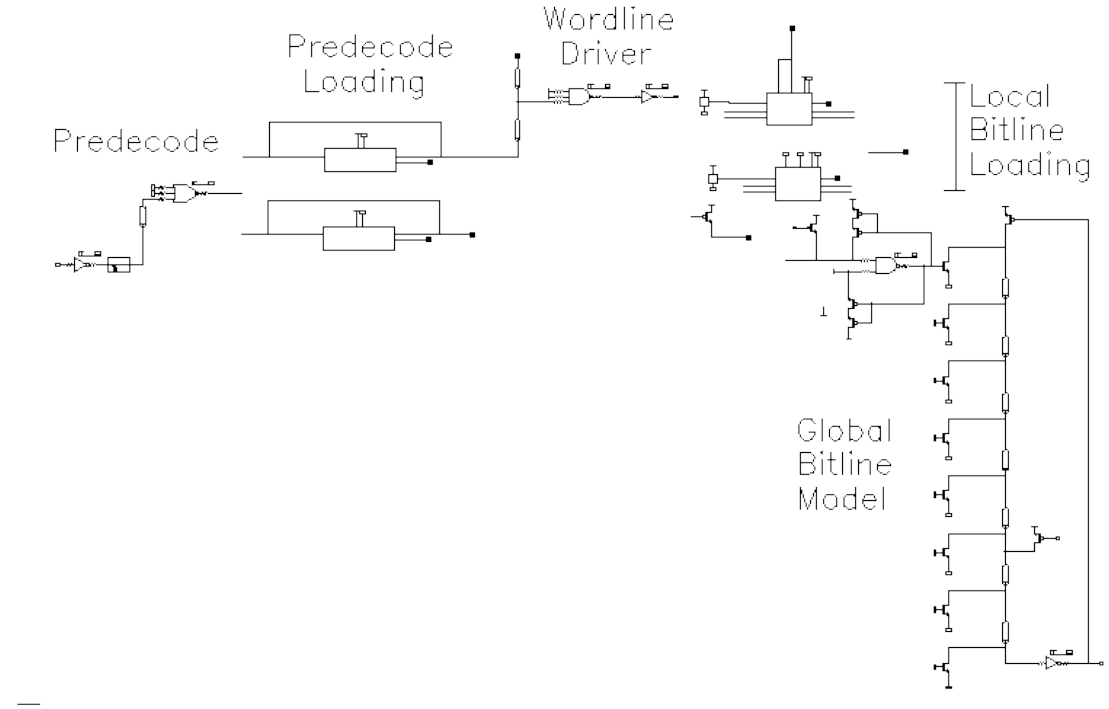


Figure A.3: Simulated cross section for partial swing design



A.4: Simulated cross section for full swing design

Appendix B

SPICE Code

B.1 Power measurement statements using current to calculate instantaneous power, average power, and energy values

```
powerInst_glob_bl = (ieglob_bl * pvdd)
powerAvg_glob_bl = (AVEVAL(powerInst_glob_bl, 0, 1e35))
pEnergy_glob_bl = (AREA(powerAvg_glob_bl, 0, 1e35))
```

B.2 dotMeasure statements use to extract maximum average power values from output waveforms

```
.meas tran POWERAVG_GLOB_BL max POWERAVG_GLOB_BL from=8e-9 to=10e-9
.meas tran POWERAVG_GLOB_PC_DRV max POWERAVG_GLOB_PC_DRV from=8e-9 to=10e-9
.meas tran POWERAVG_INPUT_INV max POWERAVG_INPUT_INV from=8e-9 to=10e-9
.meas tran POWERAVG_JNAND max POWERAVG_JNAND from=8e-9 to=10e-9
.meas tran POWERAVG_LOC_PC max POWERAVG_LOC_PC from=8e-9 to=10e-9
.meas tran POWERAVG_LOC_PC_DRV max POWERAVG_LOC_PC_DRV from=8e-9 to=10e-9
.meas tran POWERAVG_NAND_DEC max POWERAVG_NAND_DEC from=8e-9 to=10e-9
.meas tran POWERAVG_PREDEC max POWERAVG_PREDEC from=8e-9 to=10e-9
.meas tran POWERAVG_CELL max POWERAVG_CELL from=8e-9 to=10e-9
```

Bibliography

[1] Fatih Hamzaoglu *et al*., 'A 153Mb-SRAM Design with Dynamic Stability Enhancement and Leakage Reduction in 45nm High-K Metal-Gate CMOS Technology,' in *IEEE ISSCC* 2008, pp. 376-377.

[2] Harold Pilo *et al*., 'A 450ps Access-Time SRAM Macro in 45nm SOI Featuring a Two-Stage Sensing-Scheme and Dynamic Power Management,' in *IEEE ISSCC* 2008, pp. 378 – 379.

[3] Hideo Akiyoshi *et al*., 'A 320ps Access, 3GHz Cycle, 144Kb SRAM Macro in 90nm CMOS Technology Using an All-stage Reset Control Signal Generator,' in *IEEE ISSCC 2003*, paper 26.2

[4] Barbara A. Chappell *et al*., 'Fast CMOS ECL Receivers with 100-mV Worst-case Sensitivity,' in *IEEE ISSCC 1988*, pp. 59-67.

[5] Yibin Ye *et al*., 'Evaluation of Differential vs. Single-Ended Sensing and Asymmetric Cells in 90nm Logic Technology for On-Chip Caches,' *IEEE ISCAS* 2006, pp. 963-966.

[6] Kevin Zhang *et al.*, 'SRAM Design on 65-nm CMOS Technology with Dynamic Sleep Transistor for Leakage Reduction,' *IEEE ISSCC* 2005, pp. 895-901.

[7] Kevin Zhang *et al.*, 'The scaling of data sensing schemes for high speed cache design in sub-0.18 μm technologies,' *Symposium on VLSI Circuits* 2000, pp. 226-227.

[8] N. Weste and D. Harris, 'Array Subsystems,' in *CMOS VLSI Design*, pp.715-720.

[9] Anthony Aipperspach, 'SOI Sense Amplifier Method and Apparatus,' *United States Patent #6833737* 2003, pp. 1.

[10] David Brooks et al., ‘Computing Hardware,’ *Harvard University Lecture Notes – CS 141*, 2008, pp. 21.

[11] Jeff Brown., ‘Memory Hierarchy: Caching,’ *UCSD Lecture Notes – CSE 141*, 2006, pp. 13.

[12] Reza Navid, ‘Amplitude and Phase Noise in Modern CMOS Circuits,’ *Doctor of Philosophy Dissertation – Stanford University Department of Electrical Engineering*, 2006, pp. 5.

[13] S. Narendra, V. De, D. Antoniadis, A. Chandrakasan, and S. Borkar, "Scaling of Stack E ect and its Application for Leakage Reduction," in *ISLPED '01: Proceedings of the 2001 International Symposium on Low Power Electronics and Design*, pp. 195-200.

[14] B. Mohammad, M. Saint-Laurent, P. Bassett, and J. Abraham, “Cache Design for Low Power and High Yield,” in *9th International Symposium on Quality Electronic Design, 2008*, pp 103-107.

Vita

Bao Truong graduated from The University of Texas with a B.S. in Electrical Engineering in 1999. After joining IBM-Austin, he worked in packaging analysis and I/O design specializing in high speed interconnects and signal integrity analysis. Bao now works in the area of SRAM design and power analysis for IBM's z-Series server products.