

Project Report EE313, Winter 2008/09

Names: Krishna Teja Malladi/ktej/05582221
Piyush keshri/piyushk/05594497

Design Objectives: To Design Energy efficient low delay SRAM.

1. **Summary of what ideas you explored. Please list the parts of the memory that you worked on for the project, and what you tried to do. E.g. (*replace the italics text*)**

Decoder:

We used marginal cost analysis to obtain 68.7% lower power in decoder while increasing delay only by ~9%. Post charge gates have been included in the dynamic inverter as it the largest dynamic gate to save clock power. Word pulse has been also used to reset the wordline after bitlines have a sufficient divergence of 150mV. Also to reduce predecode power, we used pulse-mode half Vdd gates. The last stage of the decoder has been changed to a NOR gate to accommodate word line pulsing signal without adding extra stages on decoder critical path.

Array: The 256*256 array has been re-organized into two 128*256 banks to use divided bit lines which proved beneficial for delay by ~2x (The necessary additional muxing circuitry has been included) .

Also, the read precharge voltage has been reduced to 0.7Vdd to save Memory Core Power by ~30%. The equalizing and the precharging transistors have been sized down during the read operation to reduce clock (blpc_b) power further. To account for write operations, write precharge circuit has been designed in parallel to read which gets activated during writes, hence do not affect writes.

SenseAmp: We also tried to increase the sizings of the sense amp tail transistor to enhance the transconductance of the sense amplifier and hence, lower regeneration time. This resulted in decrease of the delay by ~11ps. However, it resulted in a huge Sense Amp power expense (24x) and there are 64 such sense amps.

TimingGen: Replica Bitline design has been used to generate ALL the required timing signals (sapc_b, sae, sel_b0, blpc_b, wl_pulse) used in various sections of the SRAM. We have assumed that the address lines and clock signal are the only signals available as external signals.

Multi Vdd: We explored the option of driving senseamp and read precharge circuitry at a low Vdd of 0.7V and were able to reduce Sense amp power by 41.9% while the delay penalty was ~35%.

2. Performance Results (give results in both ps, and FO4 delays FO4=24ps)

Clk (or address) -> WL rise	<u>139.5ps, 5.8125 FO4</u>
WL rise -> Bit Line split	<u>126.4ps, 5.26 FO4</u>
Bit Line -> SenseAmp Out	<u>73ps, 3.04 FO4</u>
SenseAmp Out -> Global Out	<u>N/A</u> (Only for divided WL)
Total	<u>338.9ps, 14.12 FO4</u>

Min Cycle Time 0.7055ns

What constrained the minimum cycle time? Which phase was critical (e.g. bitline precharge, SAE evaluation, etc)?

In the proposed design we are using blpc_b signal directly clock which made the delay of bit line precharge is most critical to our design,. So the minimum cycle is determined by the bitline precharge. However, since we are implementing wordline pulsing the bitline precharge can be done in parallel to the SA evaluation, after the wordline resets. This can reduce the minimum cycle time by ~100ps, to accommodate higher frequency clocks. Also, please note that we reduced the precharge sizings to save power here, which further increased the min. cycle time

3. Power @ $f_{clk}=1\text{GHz}$:

Decoder:	Gates: <u>0.57mW</u>	Clk: <u>0.3348mW</u>	Leakage: <u>0.12mW</u>
Memory Core:	Gates: <u>2.2057mW</u>	Clk: <u>0.422mW</u>	Leakage: <u>1.14mW</u>
Sense Ckt:	Gates: <u>0.59mW</u>	Clk: <u>0.095mW</u>	Leakage: <u>0.35mW</u>
Final Bus Drive	Gates: <u>N/A</u>	Clk: <u>N/A</u>	Leakage: <u>N/A</u>
(Only have this if you used divided WL)			
Total:	Gates: <u>3.3657mW</u>	Clk: <u>0.8518mW</u>	Leakage: <u>1.61mW</u>

Overhead Power due to timing generation circuit is $P_{\text{overhead}} = \sim 1.473 \text{ mW}$ (35% extra)

Overhead Leakage Power, $P_{\text{leakage}} = 0.17 \text{ mw}$

Total Power Consumption of Design, $P_{\text{Total}} = 7.4735 \text{ mW}$

Explain how power scales with f_{clk} .

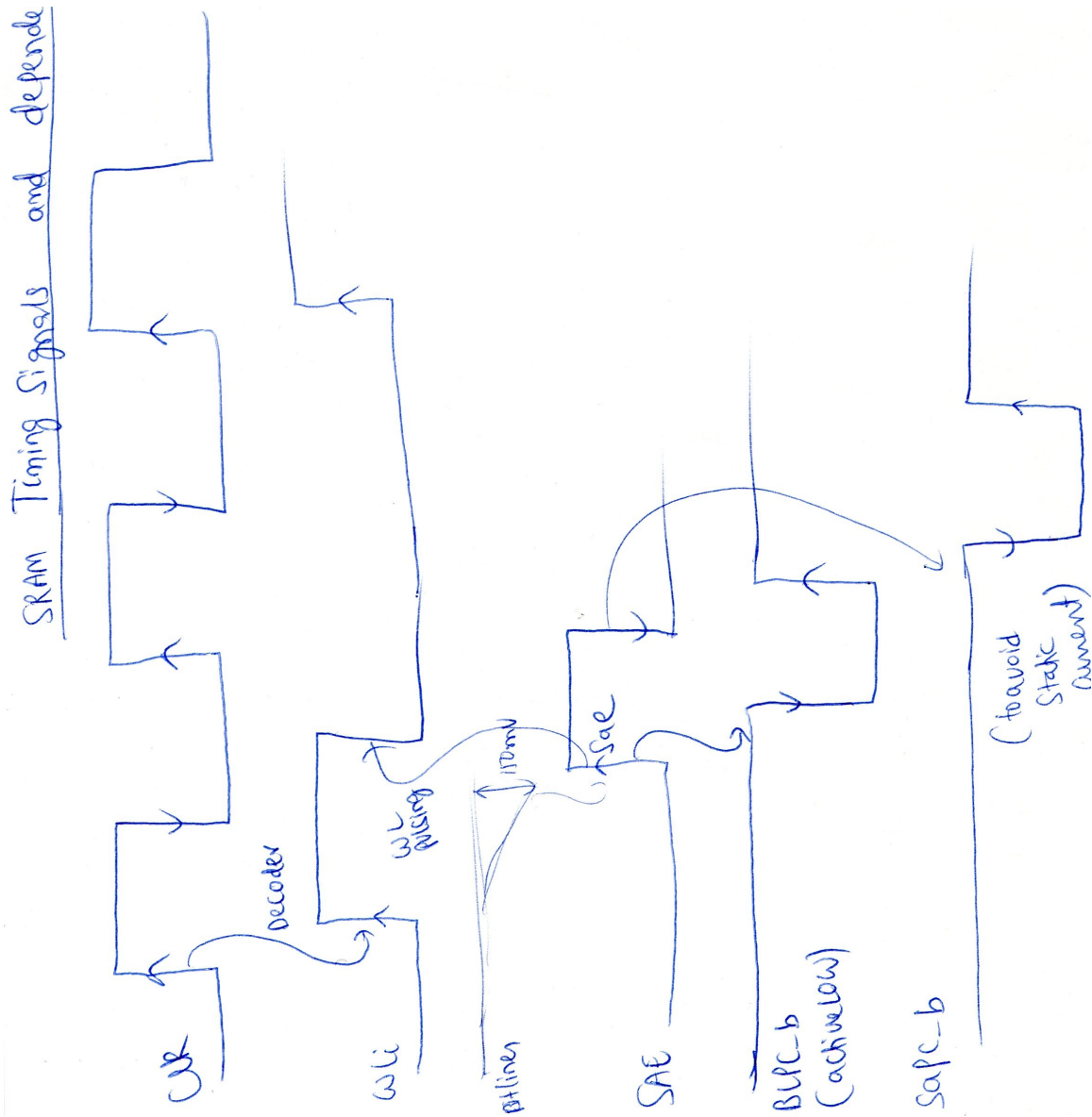
We would expect the dynamic power to scale linearly with f_{clk} .

4. Overall Timing Diagram:

Draw all the major clocks you generate (SAE, PC, SAPC, etc) and show with arrows which signals cause each transition. Also show on this diagram what limits the max cycle time.



NOTE: Please refer to the attached figure for Timing signal clocks' generation and dependencies. If bitline divergence along with its precharge back to V_{dd} are considered into a single operation, this will limit the clock cycle.

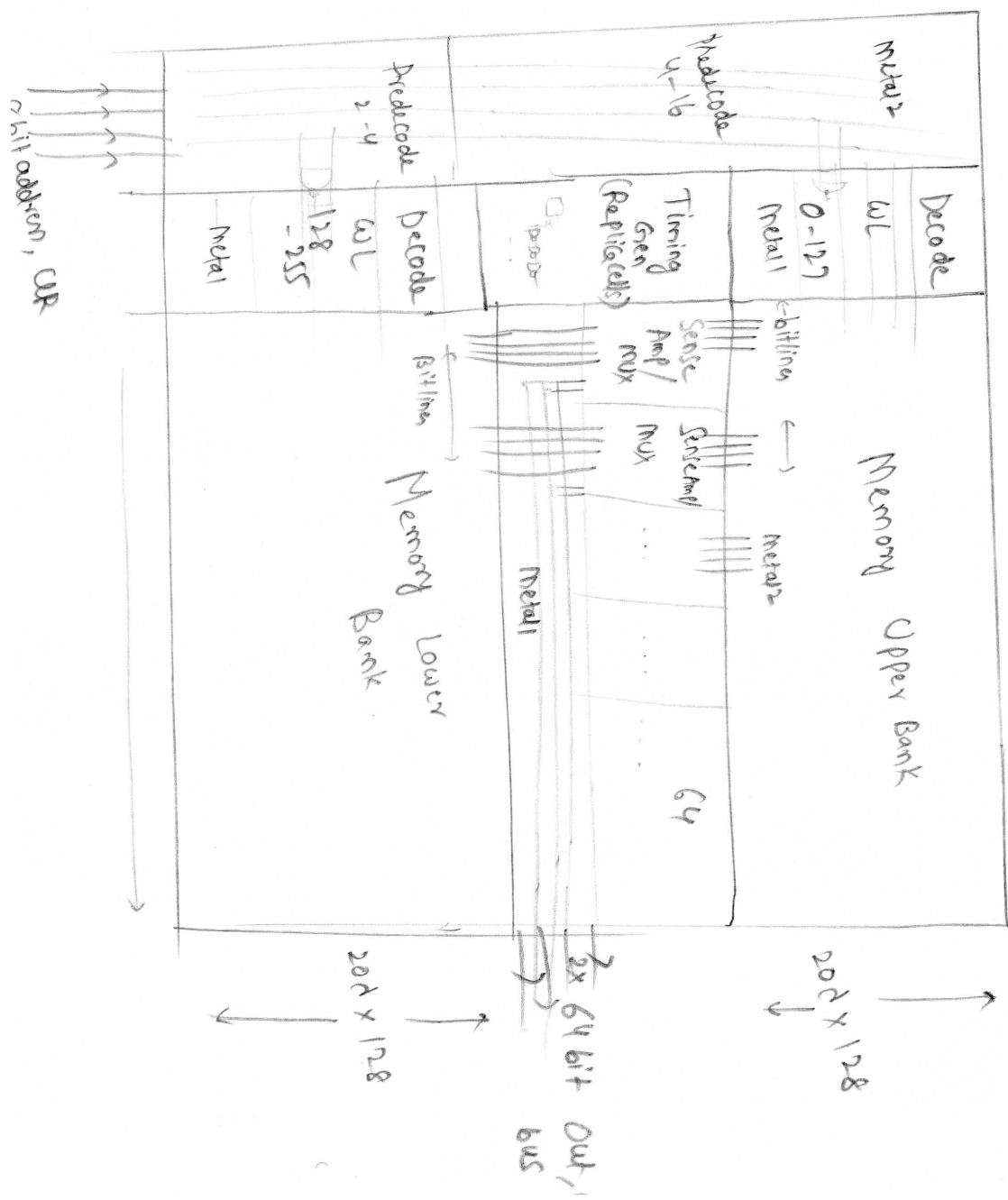


5. Basic Floorplan:

Please draw a floorplan of the chip indicating where all the parts are located. Rough size estimates for the memory (at least the memory array).

NOTE: Please refer to the attached figure for a rough floor plan. The sense-amps have been placed in between the divided wordlines to reduce the wirecap on the upper half of bitlines.

SRAM 1-600 Plan



6. Design Description

In this section, please go through each of your design changes. Each change gets its own section. For each change, please start the description with what you hoped the change would accomplish, and then describe what it actually accomplished. The last section should be description of how you implemented this change, and the performance and power of your solution. It is in these sections that I should be able to “check” the summary numbers in the front. I am more interested in your thinking and methodology than checking each gate size, so remember to talk about how you modeled wires cap, and any other issues you need to consider. If you tried multiple options, in the detailed description please mention other options that were tried, and why they were not selected.

Please also describe how you generated the different clock/control signals (like PC, SAPC, SAE) that you needed. For each signal please explain whether it is a deadly clock, the margin you gave it, and how you generated it.

6. Decoder Design

$$P_{\text{decoder}} = V_{\text{dd}}^2 * f * (32 * C_{\text{clk}} + 2 * C_{\text{predec}} + C_{\text{wordline}})$$

Approach: To reduce the overall power consumption of the decoder design, the focus was to reduce C_{clk} (having large multiplying factor), swing on predec (since, C_{predec} is large) , and Marginal cost analysis to resize the decoder for energy efficiency. Hence, the techniques implemented are as follows:

- Resize Decoder for Energy Efficiency
- Post Charge Gates
- Pulse Moder Gates
- Wordline Pulsing

6.1.1 Resizing for Energy

The base line decoder design implemented of the project was delay optimized. The marginal cost of energy for minimal delay w.r.t Energy/Power is very high. As a result, operating at minimum delay is not optimum for the design with respect to performance and power. To optimize the power consumption of the design, the delay of the decoder has been compromised by around 10% of the minimum optimized delay, which has resulted in significant power improvement (~ 70% Improvement in Power) with smaller delay cost.

Please, note that the topology of the Decoder has been changed by replacing the last inverter by a NOR gate to accommodate WL pulsing without increasing the critical decoder path.

NOTE: Kindly, see the attached circuit image.

Decoder Design

Optimal Sizing for Delay Minimization

Size	Dynamic NAND1	INV1	NAND 2	INV2	Dynamic INV3	INV4	NAND 3	INV5
$W_p (\lambda)$	8	27	11	69	258	1705	153	880
$W_n (\lambda)$	12	7	17	19	282	465	237	240

Optimal Sizing for Power Optimization

New Delay – ~10% higher than the minimum delay.

Size	Dynamic NAND1	INV1	NAND 2	INV2	Dynamic INV3	INV4	NAND 3	NOR
$W_p (\lambda)$	8	22	6	33	97	502	22	346
$W_n (\lambda)$	13	6	10	9	106	137	34	56

Power/ Performance Analysis of Decoder Design

(a) Hand Calculations

Case1: Delay Optimized

The capacitances driven by each stage in the final decoder to wordline are given by (assuming $C_g=1.4\text{fF}/\mu\text{m}=0.0308\text{fF}/\lambda$ and $C_j=1.2\text{fF}/\mu\text{m}$, $C_j/C_g=0.86$):

	Nand3	Inv4	Total
C_{load}		3218	3218
C_{gate}	1120		1120
C_{junction}	$0.86(2*153+237)$	$0.86*(880+240)$	1430
		Total	5768λ

Hence $C_{\text{WL}} = 5768\lambda = 177.65 \text{ fF}$

The capacitances driven by each stage the 4-16 predecoders are given by:

	Nand2	Inv2	Dinv1	Inv3	Total
$C_{\text{side/load}}$				731	731
C_{gate}	88	282	2170	$16*390$	8780
C_{junction}	$0.86(2*11+17)$	$0.86*88$	$0.86(282+258)$	$0.86*2170$	2440
				Total	11951λ

Hence $C_{\text{predecoder}} = 11951\lambda = 368.09\text{fF}$

$C_{\text{clk}} = (282+258) = 540\lambda = 16.63\text{fF}$

$$\begin{aligned} P_{\text{decoder}} &= V_{\text{dd}}^2 * f * 32 * C_{\text{clk}} + V_{\text{dd}}^2 * f * 2 * C_{\text{predecoder}} + V_{\text{dd}}^2 * f * C_{\text{WL}} \\ &= 1\text{GHz} * (32 * 16.63\text{fF} + 2 * 368.09\text{fF} + 177.65\text{fF}) \\ &= 0.532\text{mW} + 0.736\text{mW} + 0.177\text{mW} \\ &= \mathbf{1.445\text{mW}} \end{aligned}$$

Case 2: Energy Optimized

	Nand3	NOR	Total
C_{load}		3218	3218
C_{gate}	166		166
C_{junction}	$0.86(2*22+34)$	$0.86*(346+56)$	346
		Total	3730λ

Hence $C_{\text{WL}} = 3730\lambda = 117.5\text{fF}$

The capacitances driven by each stage the 4-16 predecoders are given by:

	Nand2	Inv2	Dinv1	Inv3	Total
$C_{\text{side/load}}$				731	731
C_{gate}	42	106	639	$16*56$	1683
C_{junction}	$0.86(2*6+10)$	$0.86*42$	$0.86(97+106)$	$0.86*639$	779
				Total	3193λ

Hence $C_{\text{predecoder}} = 3193\lambda = 98.34\text{fF}$

$C_{\text{clk}} = (97+106) = 203\lambda = 6.25\text{fF}$

$$\begin{aligned} P_{\text{decoder}} &= V_{\text{dd}}^2 * f * 32 * C_{\text{clk}} + V_{\text{dd}}^2 * f * 2 * C_{\text{predecoder}} + V_{\text{dd}}^2 * f * C_{\text{WL}} \\ &= 1\text{GHz} * (32 * 6.25\text{fF} + 2 * 98.34\text{fF} + 117.5\text{fF}) \\ &= 0.2\text{mW} + 0.197\text{mW} + 0.117\text{mW} \\ &= \mathbf{0.514\text{mW}} \end{aligned}$$

(b) Simulation Results

Decoder Stage	Power Consumption (Delay Optimized)	Power Consumption (Energy Optimized)
2-4 Predecoder	$\sim 0.01\text{mW}$	$\sim 0.0065\text{mW}$
4-16 Predecoder	$0.52\text{mW} * 2$	$0.144\text{mW} * 2$
WL Stage (Decode)	0.317mW	0.0685mW

Clock Power	0.219 mW * 4	0.0837 mW * 4
Total Power	2.24 mW	0.7 mW

Comparison in Improvement

Delay Optimized	Energy Optimized
Delay = 131 psec (Simulated)	Delay = 139.5 psec (Simulated)
$P_{\text{decoder}} = 2.24 \text{ mW}$ (Simulated)	$P_{\text{decoder}} = 0.7 \text{ mW}$ (Simulated)
$P_{\text{decoder}} = 1.445 \text{ mW}$ (Estimated)	$P_{\text{decoder}} = 0.514 \text{ mW}$ (Estimated)

Improvement in Decoder Power = 68.75 % Increase in Decoder Delay = 8.2%
--

6.1.2 Post charge Logic

The goal here is to have self reset gate for the dynamic inverter in the predecoder to reduce the capacitance seen by the clock. FO4 delay is about 25 psec. So we need a 20 FO4 delay in the buffer stages to avoid static rail current. We used the following design for the reset buffer chain for the precharge of dynamic inverter in the 4-16 predecoder to and were able to get the wordline signals while having reduced clock power. Also the 1st stage of the chain was sized at minimal $W_p=4$, $W_n=2$ to avoid loading the wordline critical path.

So with minimal delay penalty we were able to save 19.6% clock power by taking care of the inbuilt data races in post-charge.

NOTE: Kindly, see the attached circuit image.

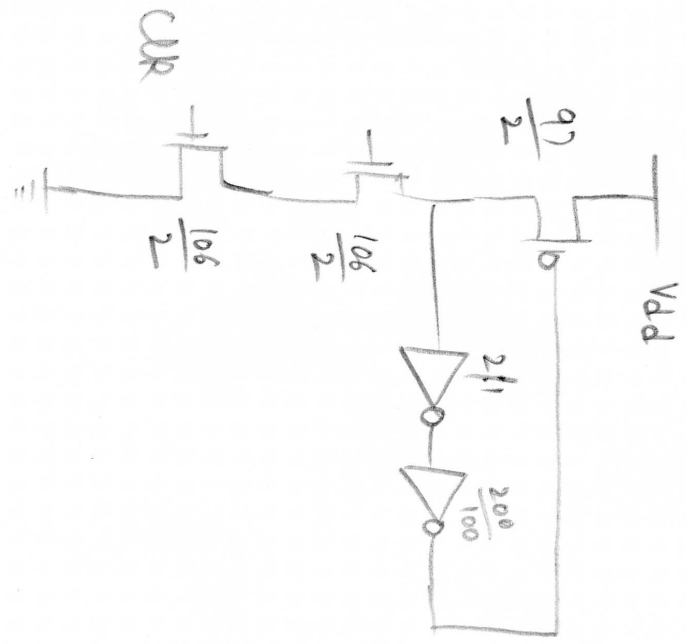
Improvements:

Before Post charge	After post Charge
Delay = 139.5 psec (Simulated)	Delay = 139.5 psec (Simulated)
$P_{\text{decoder}} = 0.7 \text{ mW}$ (Simulated)	$P_{\text{decoder}} = 0.61 \text{ mW}$ (Simulated)
$P_{\text{decoder}} = 0.508 \text{ mW}$ (Estimated)	$P_{\text{decoder}} = 0.408 \text{ mW}$ (Estimated)

Post-Charge Logic

Dynamic inverter

(5th Stage of Decoder)



(Built in Spice)

6.1.3 Pulse Mode Gates

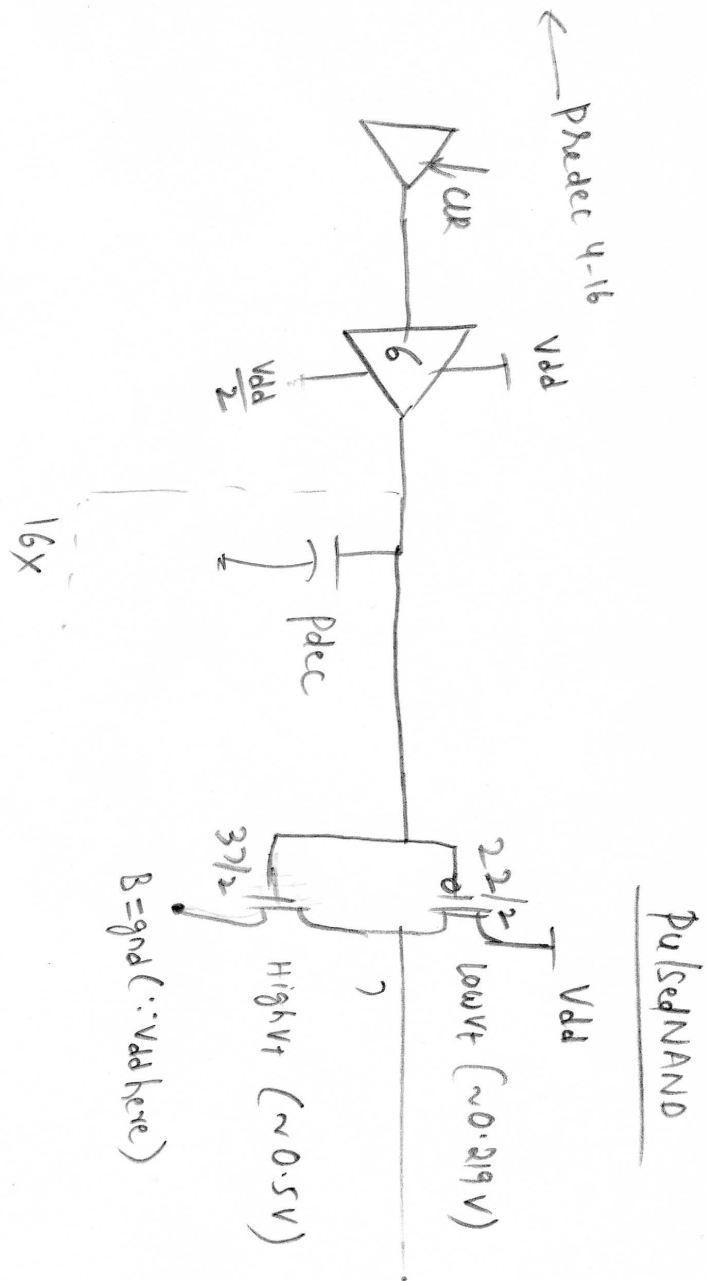
Predecode power constitutes majority of the decoder power. To reduce it by half, we used pulse mode logic gates. First, we changed the power supplies of the inverter before predecoder to Vdd, Vdd/2. This introduced noise margin issues and also slightly affected the WL falling delay (by about 4%) because of the loss of gate overdrive for the falling case. However, the swing of the predecoders has been reduced by half. To recover full swing on the wordline, we replaced the nand gate in the decode stage by pulsed NAND gate . We used a High Vth NMOS (0.5V) gate to make sure NMOS turns off when the input is Vdd/2. (Assumption: Major technologies provide nominal Vt, High vt, Low Vt transistors)

NOTE: Kindly, see the attached circuit image.

Improvements:

Before Pulse Mode	After Pulse Mode
Delay = 139.5 psec (Simulated)	Delay = 139.5 psec (Simulated)
$P_{\text{decoder}} = 0.61 \text{ mW}$ (Simulated)	$P_{\text{decoder}} = 0.57 \text{ mW}$ (Simulated)
$P_{\text{decoder}} = 0.408 \text{ mW}$ (Estimated)	$P_{\text{decoder}} = 0.309 \text{ mW}$ (Estimated)

Pulse Width Gate



(Kindly Note that we couldn't turn in Schematic as we wrote

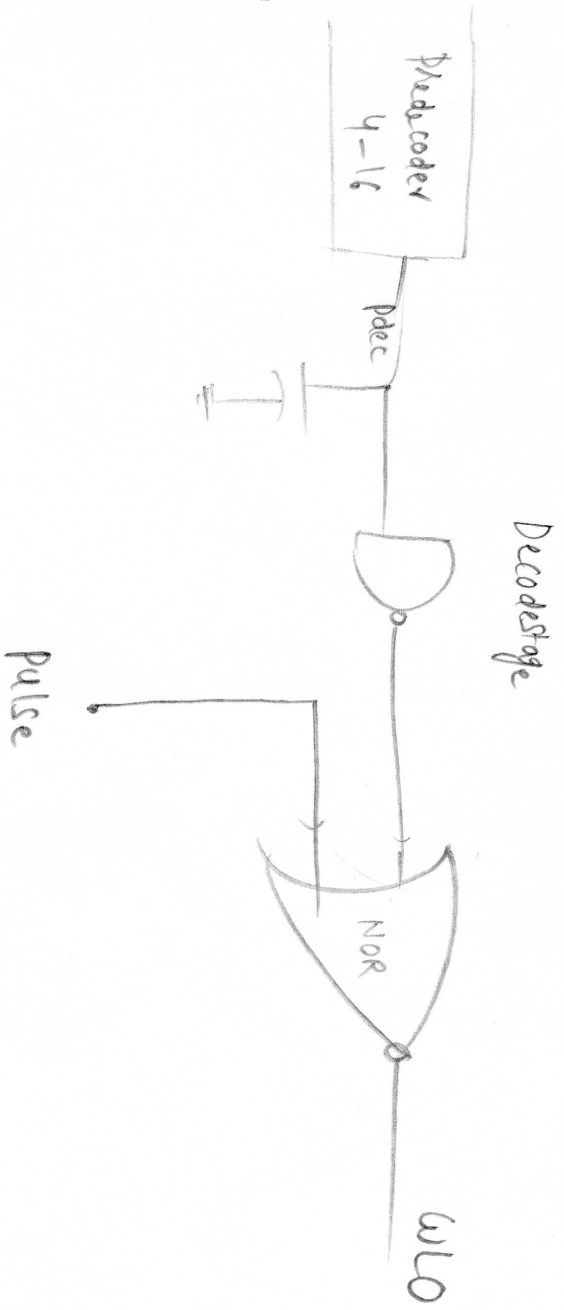
6.1.4 Pulsed word line Design

Pulsing the word line is by far the most important technique in the context of low power as the memory core power scales linearly with pulse width. But the width should be large enough for 150mV divergence on the bitlines. So the already generated SAE timing replica path has been used to generate word line gating. NOR gate has been included in the topology of the decoder design (as mentioned in 6.1.1) to achieve wordline pulsing. We have reduced the pulse mode width from (~500ps to 172ps), which results in memory core power reduction of 65.6%.

Improvements:

Before Pulse WL	After Pulse WL
BL Delay = ~132.3 psec (Simulated)	BL Delay = 132.3 psec (Simulated)
$P_{\text{mem}} = 9.16 \text{ mW}$ (Simulated)	$P_{\text{Bl}} = 3.151 \text{ mW}$ (Simulated)
% Improvement = 65.6 %	

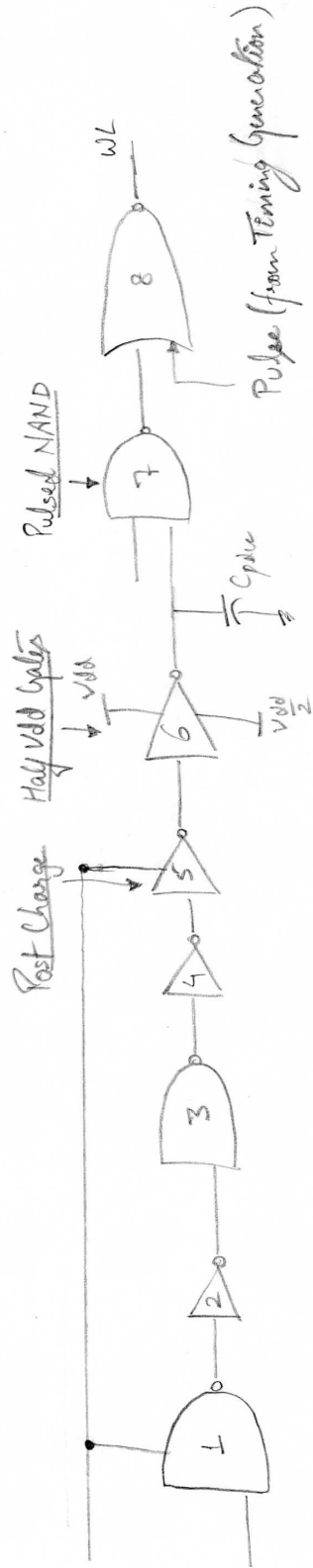
Digital Line Pulsing Using BH Replicy & NOR gate



For Pulse Signal generation, Please refer to Timing generation circuit.

Final Decoder

DECODER DESIGN



6.2 Bitlines/ memory Array:

$$P_{\text{memcore}} = 256 * I_{\text{dsat}} * V_{\text{dd}} * t(\text{wordline high}) / T$$

$$P_{\text{blpc_b}} = V_{\text{dd}}^2 * f * 256 * C_{\text{blpc_b}}$$

Approach: To reduce power we can change the factor of 256, I_{dsat} , V_{dd} and $t(\text{wordline high})$.

- I_{dsat} can be reduced by changing memory cell sizings, but results in lower write margin and hence, was not attempted.
- V_{dd} has been reduced by reducing bitline common mode voltage to $0.7V_{\text{dd}}$.
- $t(\text{wordline high})$ has been reduced by using wordline pulsing (as mentioned in section **6.1.4**).
- $P_{\text{blpc_b}}$ can be reduced by sizing down the precharge transistors.

6.2.1 Bit line common mode Voltage

i) The bitline common mode precharge voltage linearly affects the dynamic power of the memory core as it serves as the V_{dd} of the 2-stack transistors in the memory cell. The read-precharge circuit's V_{dd} has been scaled to $0.7V_{\text{dd}}$ to reduce core power. This leaves margin to make sure writes do not fail as we did not create a separate write precharge circuitry.

Improvements:

With CM voltage = 1.0V	With CM voltage = 0.7V
BL Delay = 132.3 psec (Simulated)	BL Delay = 126.4 psec (Simulated)
$P_{\text{mem}} = 3.151 \text{ mW}$ (Simulated)	$P_{\text{mem}} = 2.2057 \text{ mW}$ (Simulated)

ii) The $3*80$ lambda transistors in the read-precharge circuit can be sized down to reduce the gate capacitance to be charged during every clock cycle of 64 sense amplifiers. Doing so will reduce the precharge clock power. However, it should be ensured that the weak precharging circuit should not result in write failure. Hence, we designed separate circuitry for read and write operations. For read, we used $2*20$ lambda and a 15 lambda equalizing transistor so that the equalized voltages are within 25mV of each other. The precharge clock power has reduced linearly (~77%) while the precharge delay has increased by (~95%).

With Old precharge Sizes	With new precharge Sizes
Precharge Delay = 187.76 psec (Simulated)	Precharge Delay = 366.6 psec (Simulated)
P_{precharge} = 1.93 mW (Simulated)	P_{precharge} = 0.442 mW (Simulated)
% Improvement in Power = 77.09 %	
% Increase in Delay = 95.2 %	

6.2.2. Divided Bitlines

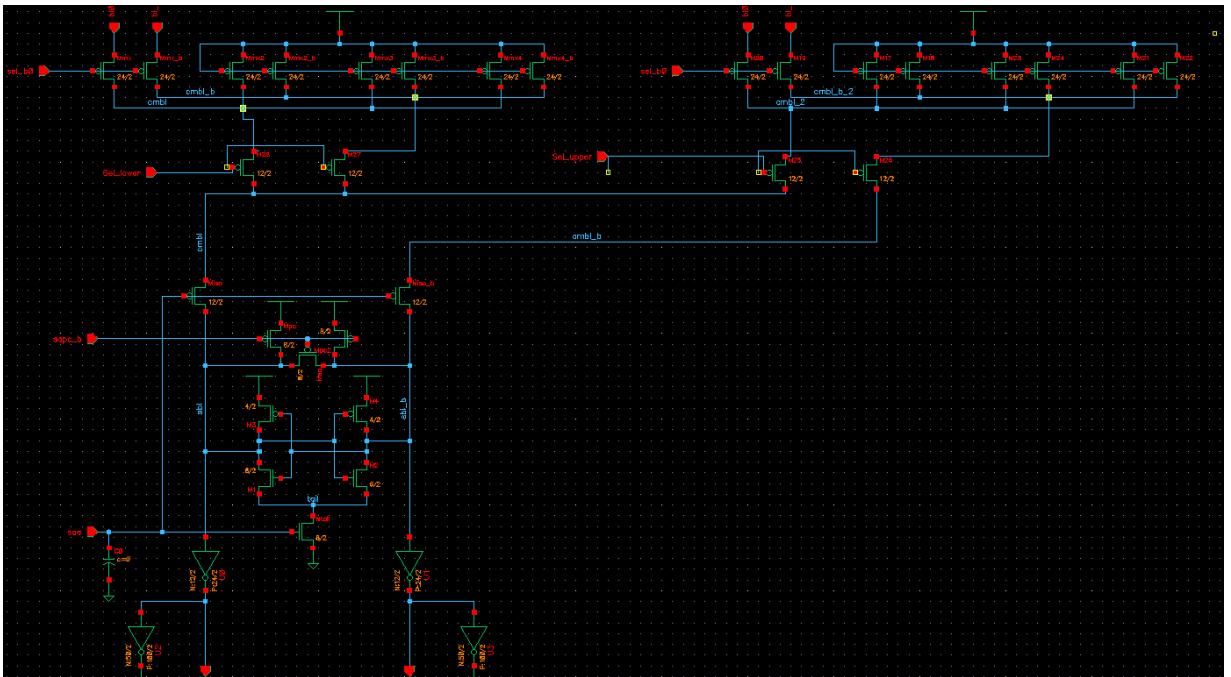
The first change to the array was to divide the bitlines into 2 banks to reduce the capacitance seen on a given bit line (~2x) which will potentially reduce power and delay of bitline fall. The 2 bank bitlines will output (cmbl_1, cmbl_b_1) and (cmbl_2, cmbl_b_2). Then, we designed 2nd hierarchy muxes which will choose between them and output (cmbl, cmbl_b) for sense amplification.

Improvements:

Before Divided Bit lines	After Divided Bit lines
BL Delay = 190.4 psec (Simulated)	BL Delay = 126.4 psec (Simulated)

NOTE: The technique has resulted in improvement less than 2x possibly because of the change in slope of wordline (using newer topology) and non-linear discharging of capacitance.

Schematic of Divided Bit Lines and MUXING



6. Power performance trade offs

$$P_{\text{sense}} = V_{\text{dd}}^2 * f * 64 * (C_{\text{switch}} + C_{\text{sae}} + C_{\text{sapc_b}})$$

Approach: The delay of the Sense amplifier is smallest as compared to wordline and bitline delays. Hence, we can use low Vdd to reduce the power at the expense of some % of delay.

6.3.1 Multi Vdd

One simple option was to have run the whole SRAM at 0.8V, which would have greatly reduced power, but we did not want to hurt delay. So, with the availability of Multi-Vdd, we explored low Vdd in different stages. We ran the Sense-Amp at low Vdd (0.7Vdd) as its delay as compared to other stages is small (even though regeneration time is exponentially dependent on Vdd). This includes sense-amp precharge too, which is not delay critical.

Bit line precharge has been implemented at Low-Vdd during the read operation as it saves power and also not delay critical. Low Vdd in memory cell will compromise writes (can be removed using Vdd for read and write separately) even though it can reduce power. But to do this, we need level converter, which can be realized as follows: (Power overhead of level converter has been added separately).

Improvements:

Sense Amps at 1.0V	Sense Amps at 0.7V
Senseamp Delay = 54ps (Simulated)	Senseamp Delay = 73ps (Simulated)
P_{sense} = 1.18mW (Simulated)	P_{sense} = 0.685 mW (Simulated)
% Power Improvement = 41.9%	

Timing Generation

Approach: All the signals required for the complete design has been generated using the basic external signals being provided (clk and address).

- Bitline replica design has been implemented to match bitline delay of the cell irrespective of process and temperature variations. This helps in creating reliable generation of SAE and other signals.
- Bitline replica design has been also been used in the design to generate pulse signal to perform wordline pulsing by connecting it with the wordline decoder and hence, helps in significant power improvement. Inverter chain has been created using bitline replica cell to precisely generate wordline pulse signal to account for the varying bitline delays during the operation of the memory and hence, to avoid any wrong read operation. Chain of inverters has been used to match the delay of the wordline (i.e. decoder) and to generate SAE signal.
- Sel_b0 signal has been generated as a short signal from the bit replica design and inverter chain to create short aperture time to load bitline data for SA evaluation. SAE signal is low during the loading time of data to sbl and sbl_b0 and goes high only after Sel_b0 signal goes high.
- Sopc_b signal has been generated from the inverter chain (derived from bitline replica) to create high pulse during evaluation period which goes low after the bitline replica switches.

NOTE: 30% delay has been added while designing the SAE signal to account for the timing margin (to ensure that correct data is read by the sense amplifier even if the clock arrives earlier than the desired time). Moreover, the power supply connected to the timing generation circuit is separate from the rest of the memory cell. Hence, regulated power supply can be connected to the timing generation circuit to account for process variations and timing jitter can be reduced to ~10%.

Method: Since, the chain of inverters has been used to match the delay of the wordline and various signals has been derived to generate the SAE signal precisely; the various side loads come into play and sizing becomes an issue. Careful sizing of the inverters has been to minimize the power consumption of the timing generation circuit. Fanout per stage is higher than 4 (~6) to account for lesser power for smaller delay.

Load on SAE: $32\lambda * 64$ (Sense Amplifiers) * $C_g = 63\text{fF}$

Load due to pulse NOR (decoder) = 9.3 fF

Load because of sel_b0 = 67.8fF

Load due to sapc_b= 20 fF

NOTE: Kindly, see the attached circuit image.

Now, the load of inverter chain on the replicated bitline needs to be small, the input load on the inverter chain has been kept minimum. As the number of stages of the inverter chain increases, it results in the decrease of the fanout per stage (f). However, as the number of stages increases, the power consumption of the inverter chain goes up. But, if the number of stages decreases, the fanout per stage (f) increases (beyond ~4) which in turn increases the delay of the signal (not desired – results in slow response and increase aperture and sensing delay). However, it results in lower power consumption.

To take this trade off into account the inverter chain has been optimized to a certain fanout (f) range (2-6).

Bitline Replica Design

Objective: Bitline replica design has been created to match the delay of the bitline during read and to create pulse wordline so that the power consumption in the wordline can be reduced after the data stored in the memory have been read and send to the sense amplifier. The delay of the replica design has been matched to delay of the bitlines.

Simulation Results

We instantiated a total of 11 dummy cells to match the delay of a bit line going low to 850mV.

Delay of replica design (from Vdd – vdd/2) = **~121.6 ps**

Delay of bitline (from vdd – (vdd-150mV)) = **~126 ps**

NOTE: Hand analysis matches with Simulation. Waveforms of relevant nodes are attached.

Simulation Results

The simulated results are as follows:

1. Word Line to bit line Delay= 126.4 ps

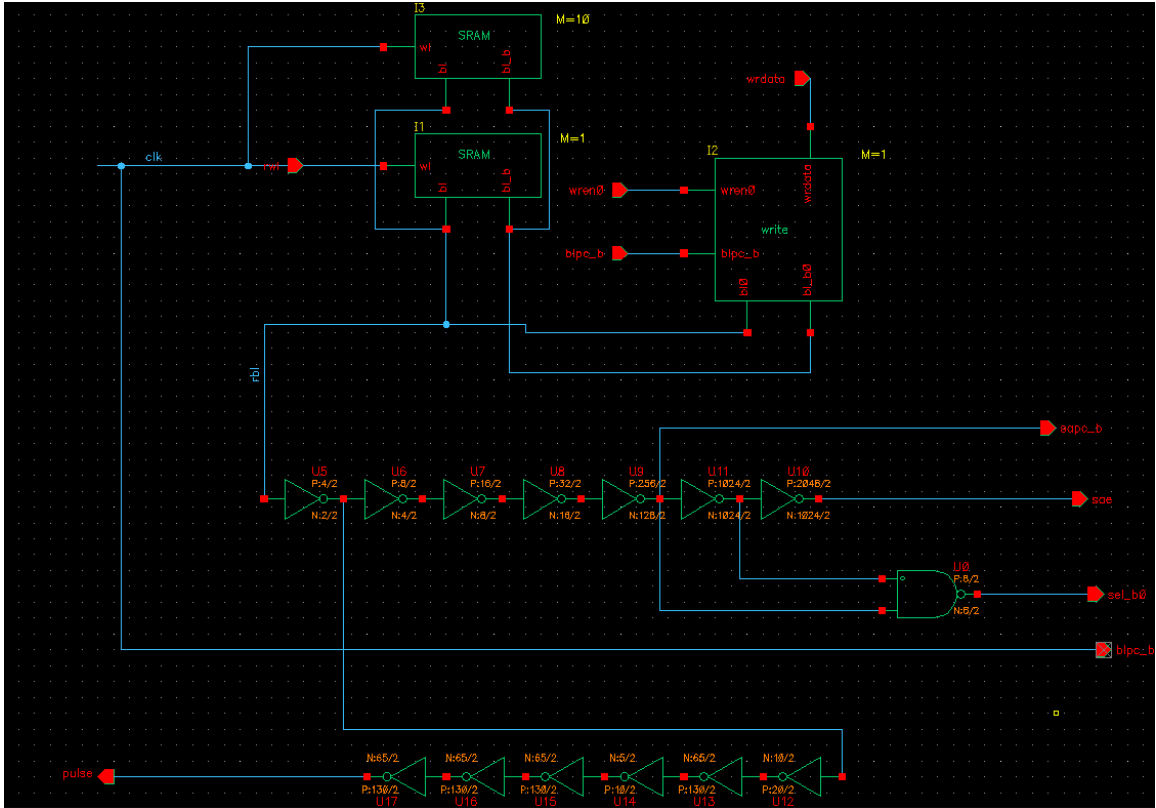
2. clk to bit line delay= Decoder delay + BLLine delay= 139.5ps + 126.4ps = 265.9 ps

3. Dummy word line to dummy bit line= ~122ps (note that this approximately matches the BL delay)

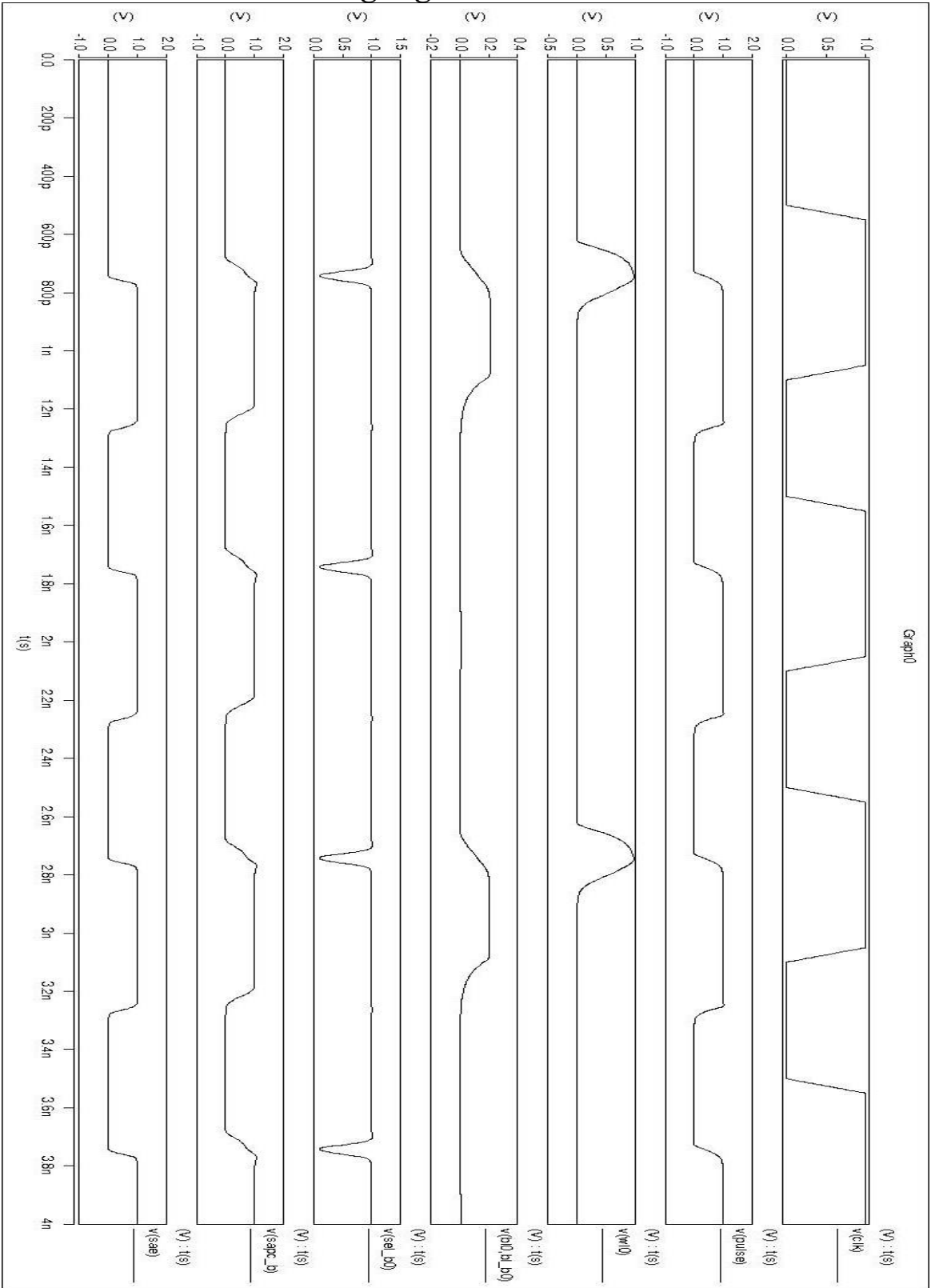
4. 30% timing margin on SAE==> We should take care of case when SAE arrives 30% late and there is not enough bit line divergence = **1.3 * 265.9 = 345.67ps**

5. Simulated clk to SAE= ~350 ps

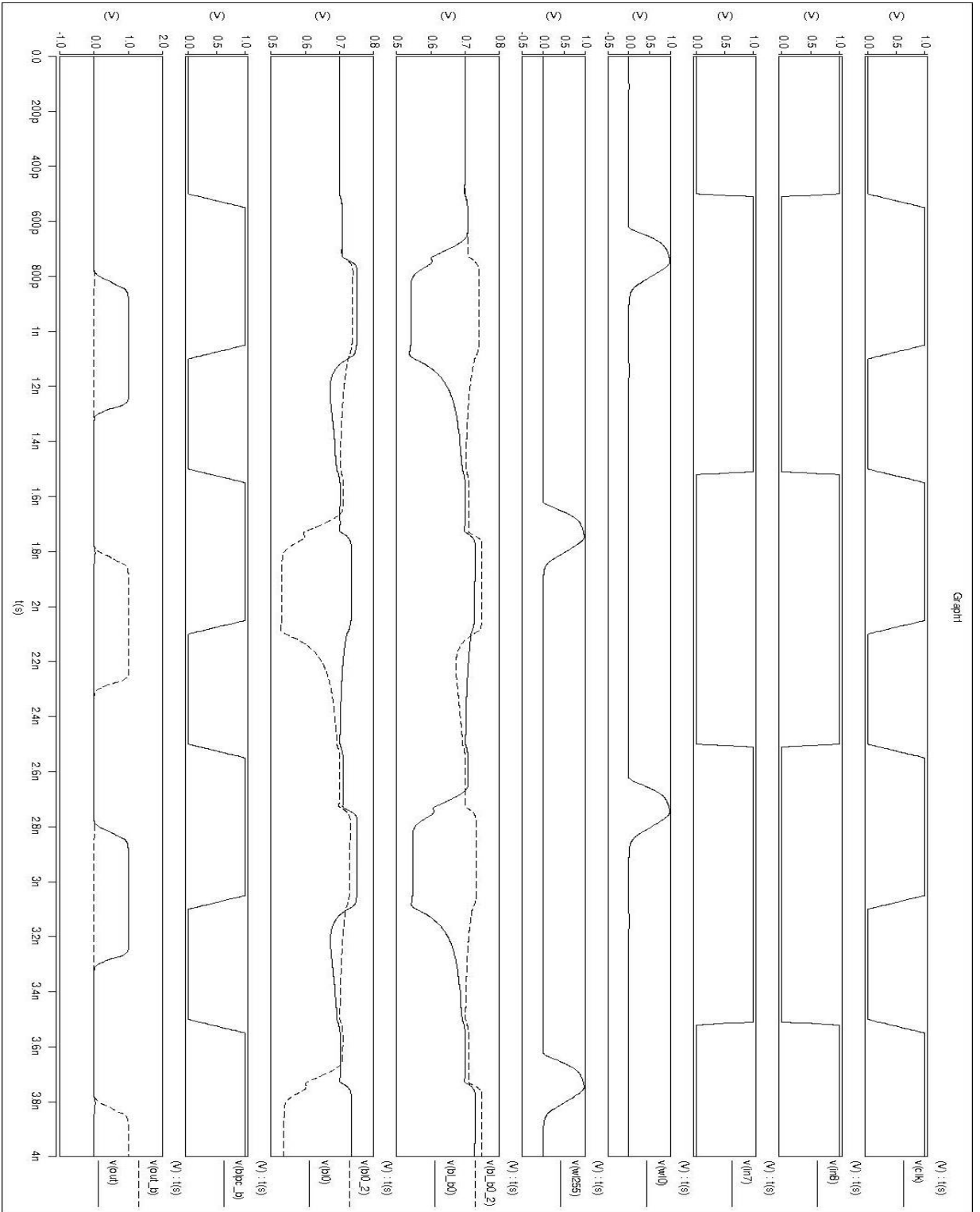
Schematic of the Timing Signal generation circuitry



Timing Signal Waveforms



Overall Simulations



7. Analysis – The memory you wished you designed

In this section you should describe the memory you wished you created. If you are very happy with you design, then use this section to explain why you like you design, and the reasons you don't want to change it. If there are sections of the design you don't like, please use this section to review the issues with you current approach, and the approach (in hindsight) you wished you had taken.

- **Pipeline Read:** Since, the wordline is active only for $\sim 270\text{ps}$ (wordline Delay + Bitline Delay = $\sim 270\text{ps}$) in the complete cycle of $\sim 705\text{ps}$, the objective is to pipeline the read process from decode stage to evaluation stage. After Wordline pulsing the wordline will fetch the other address, during which the SA will be evaluating the previous data. To achieve this we will generate the internal clock within the design with twice the frequency of the external clock. Address will be fetched at every edge of the external clock (irrespective of positive and negative). Since, the wordline and SA will be executing in parallel different address, it will help in reducing the effective clock cycle for each read. However, to achieve this the timing generation will become more overhead on the overall system, but will enhance the throughput of the design by a factor of 2. Due to lack of time, we framed out the plan but could not execute it. This implementation helps in utilizing the hardware to read more bits. (It would surely increase Power and part of the reason why we did not implement in our low-power centric design). But the control to generate so many timing signals quickly grew (unlike digital logic where we have Registers as signal barriers) and we had to drop the idea.
- **Separate Read/Write Circuitry:** We also could have generated separate read precharge and write precharge circuits which would have allowed to reduce bit line common mode voltage even further. We also tried to increase the sizings of the sense amp tail transistor to enhance the transconductance of the sense amplifier and hence, lower regeneration time. This resulted in decrease of the delay by $\sim 11\text{ps}$. However, it resulted in a huge Sense Amp power expense (24x) and there are 64 such sense amps.