第四次作业

题目一解答

- DRAM: DRAM是用电容有无电荷来表示信息0和1,为防止电容漏电而导致读取信息出错,需要周期性地给电容充电,即刷新;我们平常所提到的计算机的内存指的是动态内存(即DRAM),用来存放当前正在使用的(即执行中)的数据和程序。
- SRAM: 而SRAM是利用触发器的两个稳态来表示信息0和1,所以不需要刷新。另外,SRAM的存取速度比DRAM更高,常用作高速缓冲存储器Cache,但是成本比DRAM更高。
- ROM: ROM在系统停止供电的时候仍然可以保持数据,而RAM通常都是在掉电之后就丢失数据,典型的RAM就是计算机的内存。

题目二解答

(a)

分为差分感应放大器和与充电电路;

(b)

读操作时,位线BL, \overline{BL} 均预充电至 $\frac{V_{DD}}{2}$,实际的预充结果可能会有一点偏差,在于充电电路中会在二者之间进行电压平衡,避免当进入差分感应时被放大二导致识别错误。

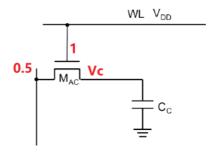
题目三解答

(a)

已知:

- 导电因子: $K_N = \frac{W}{L}K'_N$
- 考虑体效应的阈值电压: $V_T=V_{T_0}+\gamma(\sqrt{2\psi_F-V_{BS}}-\sqrt{2\psi_F})$, 其中 $V_{T_0}=V_{FB}+2\psi_F+\gamma\sqrt{2\psi_F}$

当DRAM单元内储存的是1时, 电路图如下:



此时 $V_T = 0.7 + 0.5 * (\sqrt{0.6 - V_{BS}} - \sqrt{0.6});$

存入1时, 充电过程将在 $V_C = V_{DD} - V_T$ 停止;

由于 $V_{BS} = V_B - V_C$ 且NMOS管中 $V_B = 0$;

整理得到方程: $3-V_C=0.7+0.5(\sqrt{0.6+V_C}-\sqrt{0.6})$

设 $V_C = x$

初步整理得到: $4.6-2x=\sqrt{0.6+x}-\sqrt{0.6}$

换元: $\sqrt{0.6 + x} = t$

得到: $2t^2 + t - 5.8 - \sqrt{0.6} = 0$

解得: $t_1 \approx -2.08, t_2 \approx 1.58$

舍去负解,代回解得: $V_C = 1.8972 \text{ V}$

(b)

升压字线,仅改变 $V_G=5$

即初始方程变为: $5-V_C=0.7+0.5(\sqrt{0.6+V_C}-\sqrt{0.6})$

由于只改变了常数项,所以易得最终方程为: $2t^2 + t - 9.8 - \sqrt{0.6} = 0$

解得: $t_1 \approx -2.56, t_2 \approx 2.06$

舍去负解,代回解得: $V_C = 3.6436 \text{ V}$

从结果可以看出,通过升压字线的方式,可以解决NMOS传输门充电不完全的性质。

题目四解答

(1)

由于要计算漏电流,先把考虑体效应的阈值电压求出: $V_T=V_{T_0}+\gamma(\sqrt{2\psi_F-V_{BS}}-\sqrt{2\psi_F})$

①求解i1:

$$V_{TN_1} = 0.7 + 0.5(\sqrt{0.6 - 0} - \sqrt{0.6}) = 0.7V$$

此时
$$V_{DS} = 1.5V < V_{GS} - V_{TN_1} = 3 - 0.7 = 2.3V$$

故工作于线性区,代入相应公式计算得到:

$$i_1 = 30 * [(3 - 0.7 - 0)^2 - (3 - 0.7 - 1.5)^2] = 139.5 \mu A$$

②求解i2:

$$V_{TN_2} = 0.7 + 0.5(\sqrt{0.6 + 1.5} - \sqrt{0.6}) = 1.037V$$

此时
$$V_{DS}=1.5V>V_{GS}-V_{TN_2}=0.463V$$

故工作于饱和区,采用对称形式公式计算的话后面的平方项无效:

$$i_2 = 30 * [(3 - 1.037 - 1.5)^2] = 6.431 \mu A$$

故得到
$$i_1 = 139.5 \mu A, i_2 = 6.431 \mu A$$

(2)

需要保证M2,M3两管不能导通,否则会改变SRAM CELL内部数值状态。

(3)

读操作时对被读数据D, D所造成的影响是微小的,

由于SRAM CELL内部是由两个反相器交叉耦合构成的双稳态电路,

其逻辑电平转化曲线的不理想区域使得其相当于一个高效的放大器,

所以就算被读数据D,D有稍微改变,经过内部的正反馈效应能够很快使得数据恢复到双稳定状态,

也即相当于保证了无损读取。