集成电路设计实习 VLSI Design Labs

综合实验：数字定制设计

SRAM的设计

Institute of Microelectronics Peking University

实验目的

- 本实验为“集成电路设计实习”课程综合实验之一，实验的主要目的是采用定制的设计方法，完成128×8bit的SRAM的设计
- 面向Chart 0.35um工艺，完成电路设计，电路仿真、版图设计和版图验证等设计过程
- 通过选择本实验内容，同学可以学习掌握cmos集成电路的设计方法，熟悉从电路分析，电路设计到流片和测试的设计过程

设计时间安排

- 从2011年11月23日到最后一次课
- 12月30日为设计数据提交的截止日期，提交数据，用于流片
- 12月30日也是设计检查的截止日期，根据设计完成的情况进行综合实验评分
- 合理安排课时分配
- 前仿不要占用太多的时间，尽量在2-3次课内完成，主要的工作量在版图和后仿

实验要求

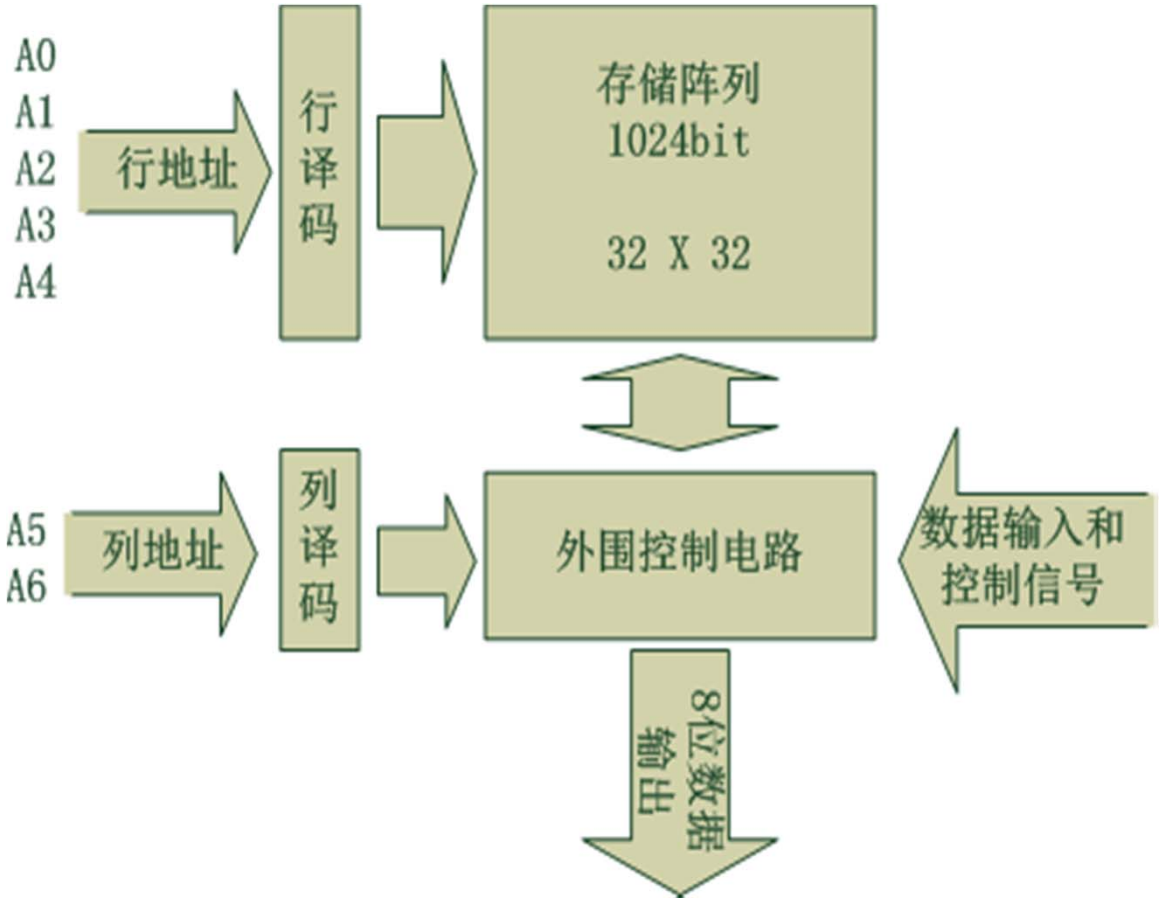
- 功能：128×8位的SRAM的设计，包括对存储单元的按字节的正确的读写操作，每一个地址都可以完成读写操作
- 速度：读写访问时间不超过2ns
- 面积：core部分面积没有要求（尽可能小），对整个芯片IO不超过28个
- 功耗：没有具体要求，尽可能的低功耗设计
- 可靠性：没有要求，能保证电路稳定工作即可

实验内容

- 电路设计
- 版图设计
- 版图验证
- 封装和测试安排（选做）
- 实验报告
- 实验陈述ppt
- 芯片测试和测试报告（选做）

体系结构

- 存储阵列：6管SRAM单元结构。划分为32行32列（正方形矩阵）。
- 译码电路：行译码和列译码。5-32行译码器输出选择一行。2-4列译码器选择四组8列数据之一输入输出。
- 外围控制电路：预充电路和数据读写控制以及灵敏放大器。
- 数据和地址缓冲器：可选。



芯片结构

- 输入信号：
  - A6－－A0，7位地址信号
  - WE_READ，读写控制信号。高电平写，低电平读。
  - PRE，预充控制信号。PRE为低时，电路预充。预充阶段不读不写，所有地址均无效。可以用来当作片选信号。
- 数据信号：
  - I/O0-I/O7，8位数据输入/输出信号
- 电源：电源和地信号
- 自己可以根据需要，添加额外的控制信号（例如译码电路和灵敏放大器的控制信号）

功能和时序描述

- 电路在PRE和WE_READ信号的控制下，进行读写操作
- 读访问时间：地址有效到数据读出的时间
- 写访问时间：地址有效到数据正确写入的时间

实验过程

- 存储单元的电路设计和仿真
- 预充电路的设计仿真
- 灵敏放大器的设计仿真
- 1列存储单元、预充电路和灵敏放大器组成模块电路的仿真
- 行列译码器的电路设计仿真
- 存储器整体电路仿真
- 存储单元的版图设计
- 根据单元版图确定灵敏放大器、预充电路的版图宽度
- 根据阵列高度和宽度确定译码器的版图结构
- 存储器整体的版图设计和验证
- 存储器版图提取和后仿真

设计内容

- 电路设计：原理图输入和spectre仿真
- 版图设计：定制版图设计，使用两层金属。版图布局不要超过2:1
- 版图验证：cadence公司dracula工具的drc和lvs检查
- 设计库：chrt的标准单元库和标准IO库

电路设计

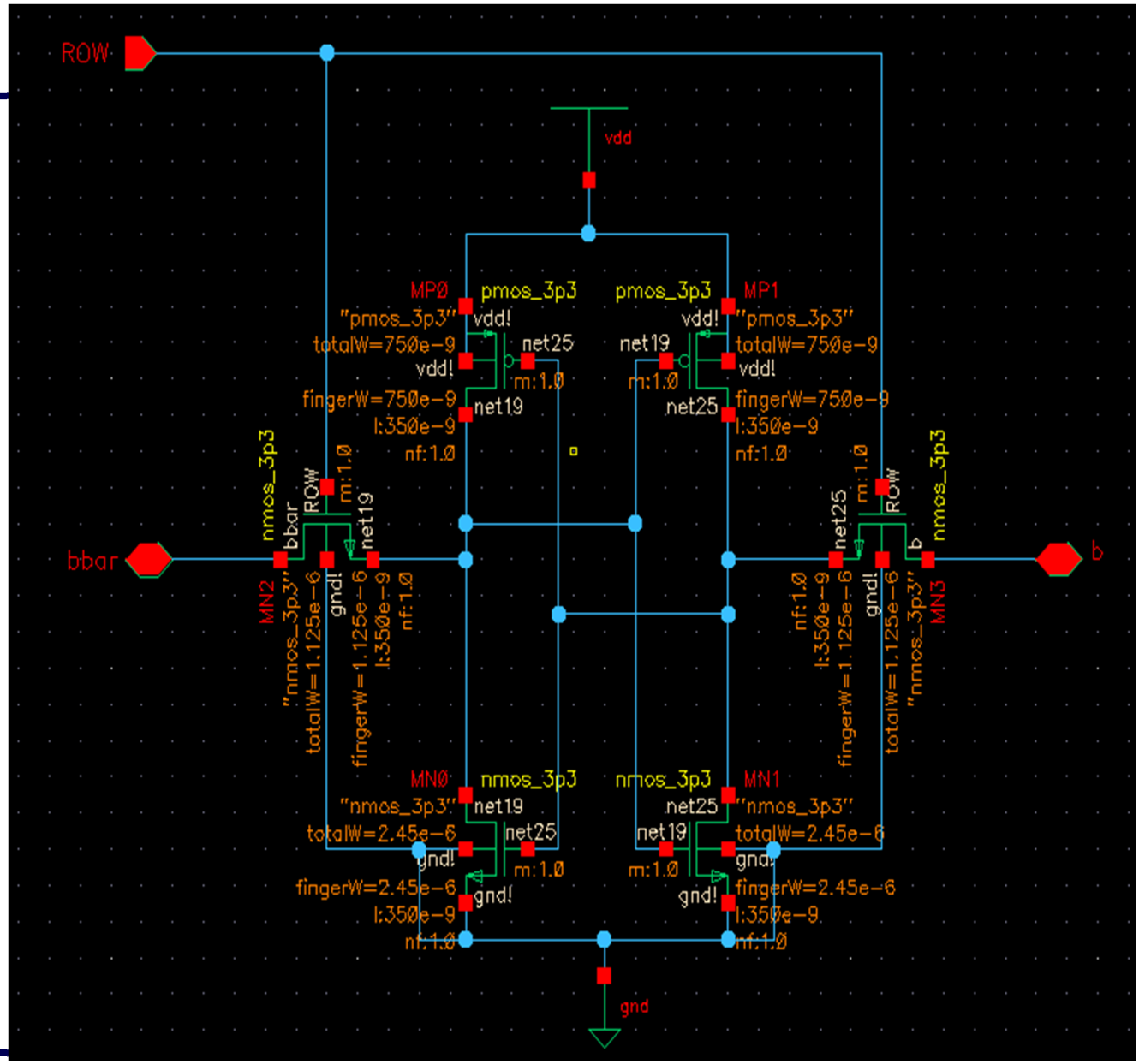
- 在开始电路设计之前，希望大家首先能清楚SRAM的工作原理。具体可以参见甘学温老师的《集成电路原理与设计》第七章的内容。
- 这次设计的SRAM主要包括下面几个小的模块：
  - 6管存储单元阵列
  - 预充电路
  - 数据输入电路
  - 灵敏放大器
  - 另外还有译码器和控制信号产生

电路设计

- 在开始画电路之前，需要提醒大家注意下面几个问题:
  - 第一：PMOS衬底接VDD，NMOS衬底接GND，不要忘记。
  - 第二：电路连线如果太复杂，建议用wire name来连接。只要将不同的线命名为相同的名字，就表示连接在一起了。
  - 第三：生成symbol的时候考虑一下端口的位置，以方便后面调用时模块间的连线。
  - 各个模块尺寸的选择可以参考讲义图中的尺寸，但这并不是最优化的尺寸，可以自行优化设计尺寸。
  - 各个模块的设计可以参考讲义中的结构，也可以自行查阅相关文献优化设计。
  - 有一点需要说明一下。大家在画电路的时候，可以使用标准单元库中的单元。但是标准单元库只有版图，所以电路图中只会有一个空的schematic。这样的话画版图快一些，但是不能通过LVS查错。只能通过后仿真来确定自己电路是否正确。大家自己选择实现方式。

六管存储单元

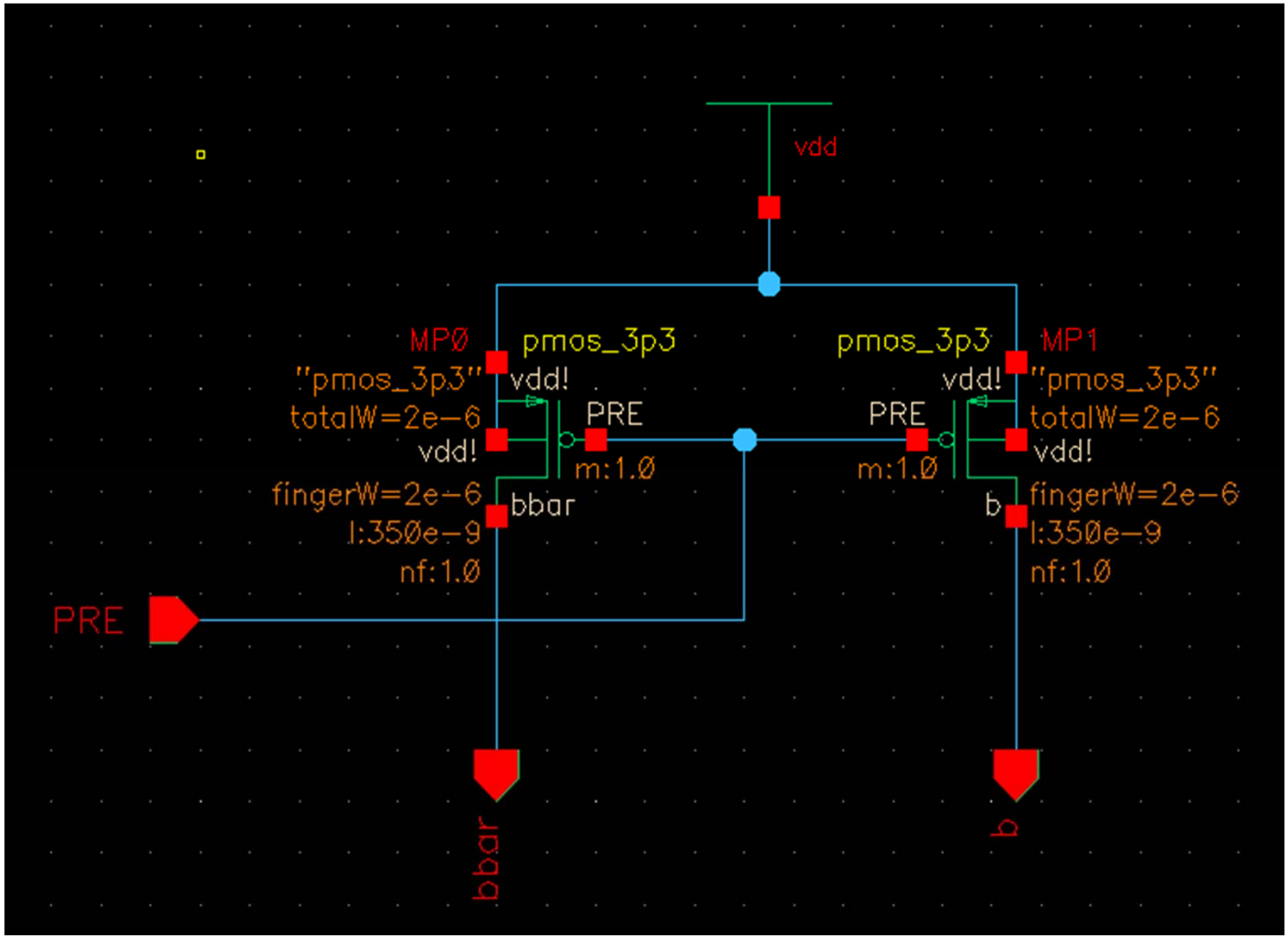
- 六管存储单元参考电路如图：
- 尺寸选取主要是要考虑数据能正常写入。就是说数据写入模块能够改写存储单元的数据。



预充电路

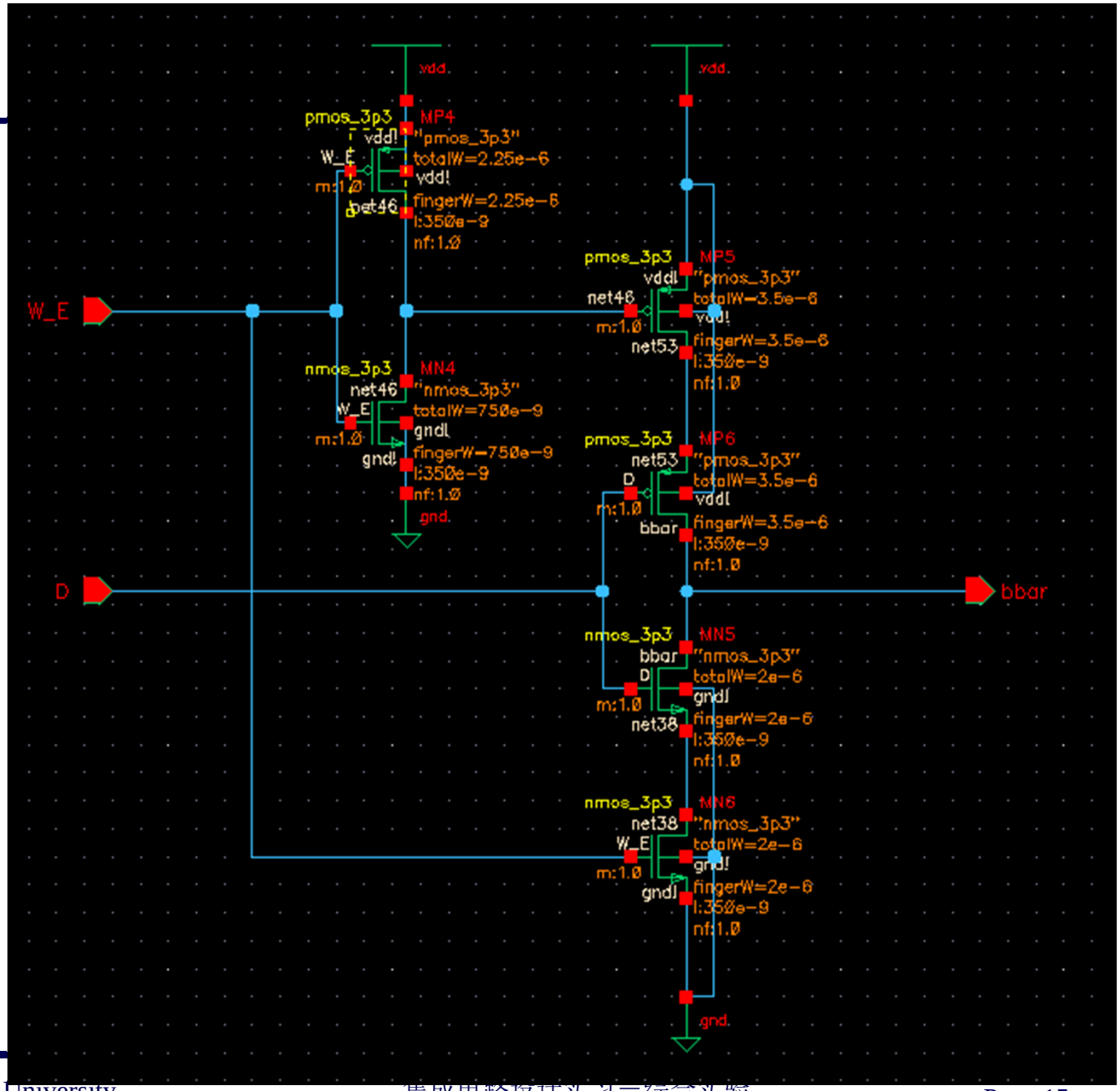
●预充电路：

预充控制信号PRE为低时，两根位线都被预充到高电平，为了保证充电的速度适当的将尺寸设计的大一点



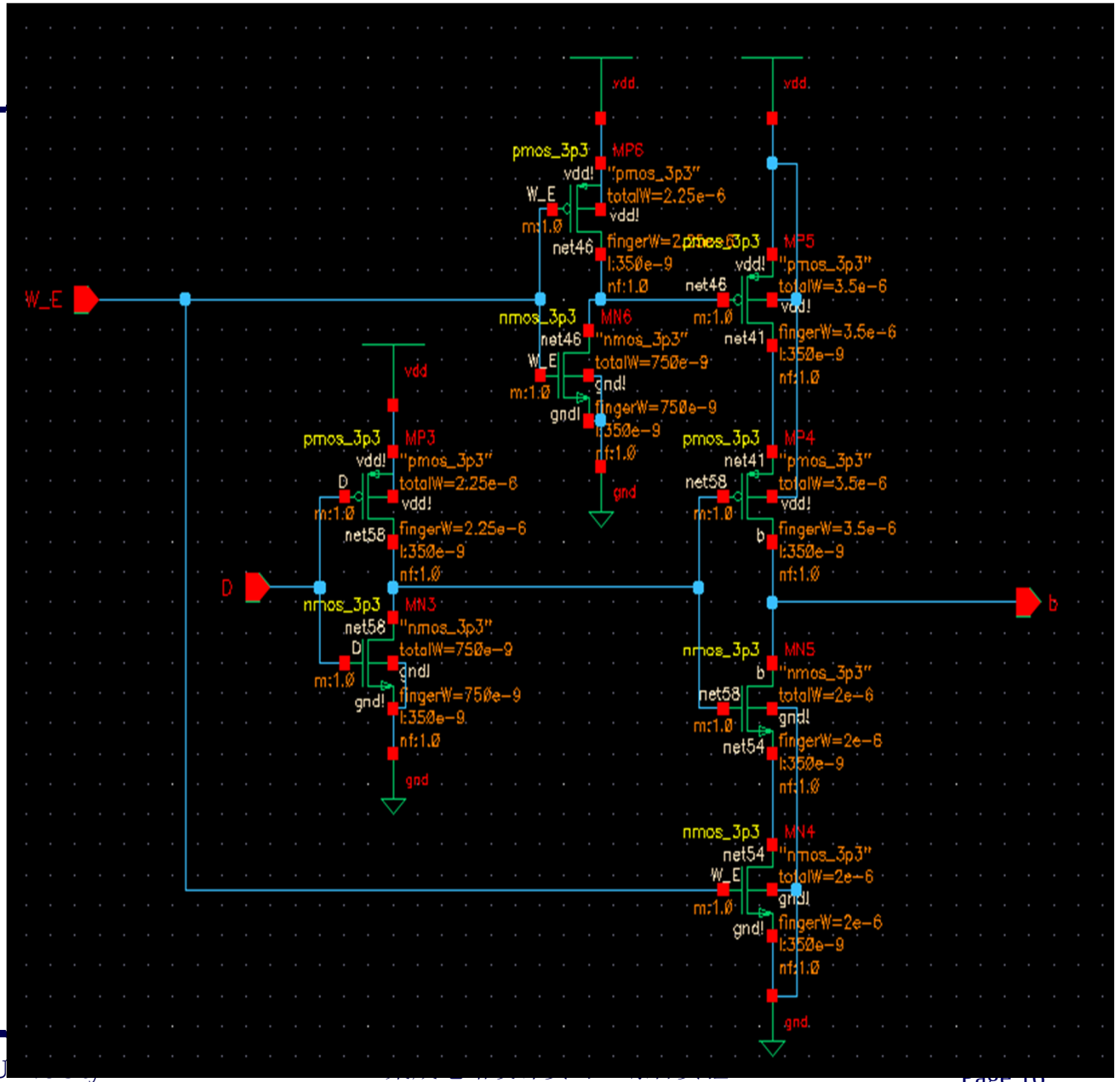
数据写入电路

- 数据写入电路1
  - W_E为低时，bbar处为高阻，数据不能写入。
  - W_E为高时，bbar为D的反信号。



数据写入电路

- 数据写入电路2
  - 就是在数据写入电路1的基础上多了一个反相器，与位线b相连
  - 数据写入电路尺寸选取，主要是要驱动能力足够改变存储单元的数据。

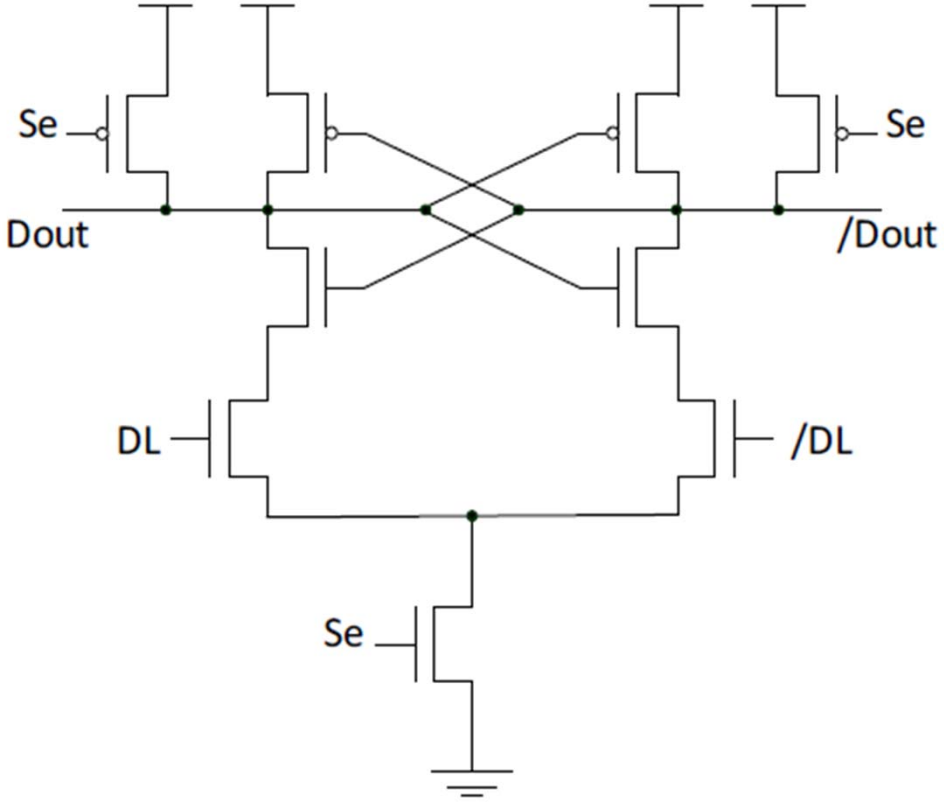


灵敏放大器

- 灵敏放大器结构如图

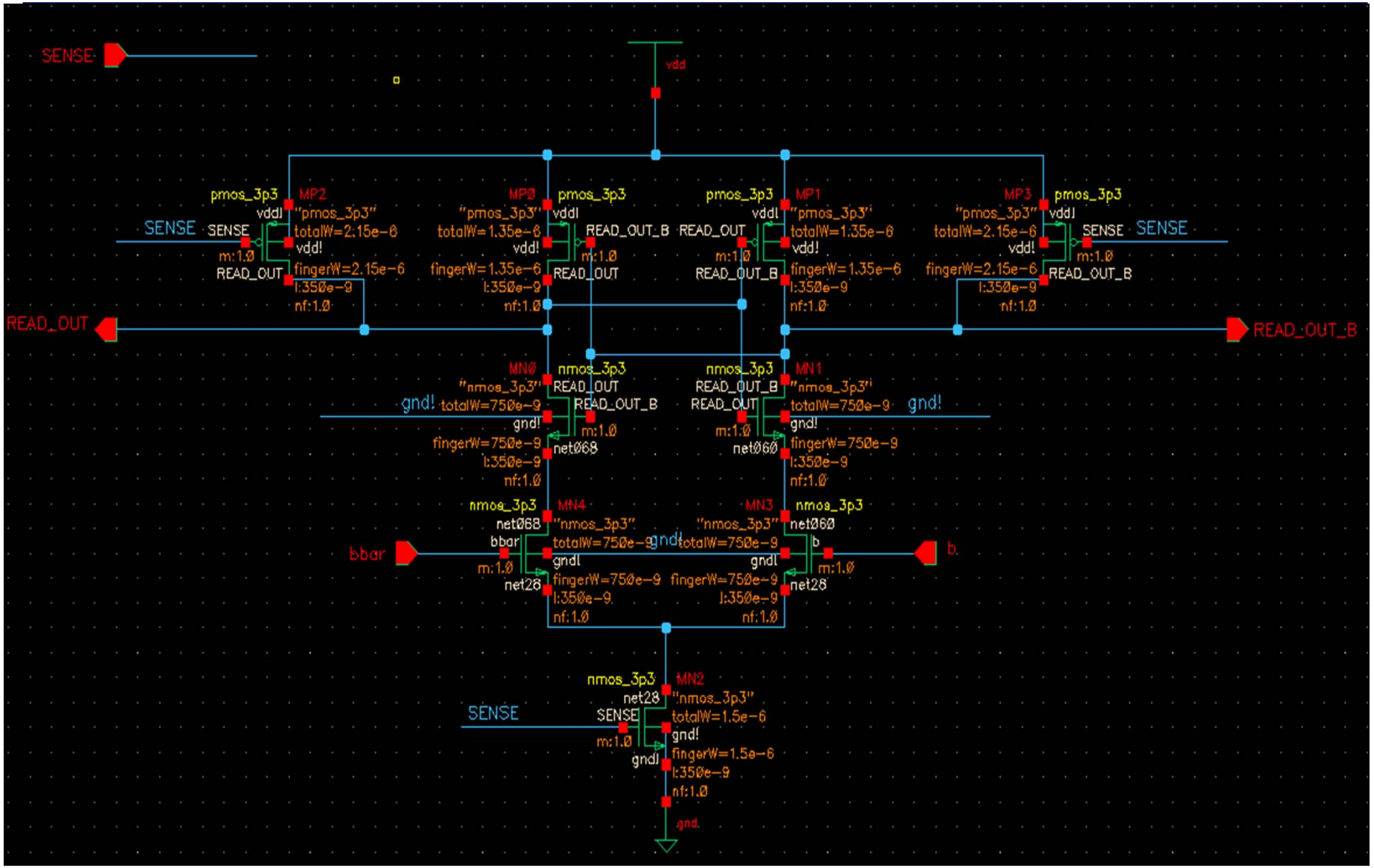
  Se为低时，Dout和/Dout预充到高。接地NMOS管断开。

  当需要读出时，Se变高，接地NMOS导通，这时会根据两根位线DL和/DL的不同，产生不同的放电电流，从而读出相应的信号。



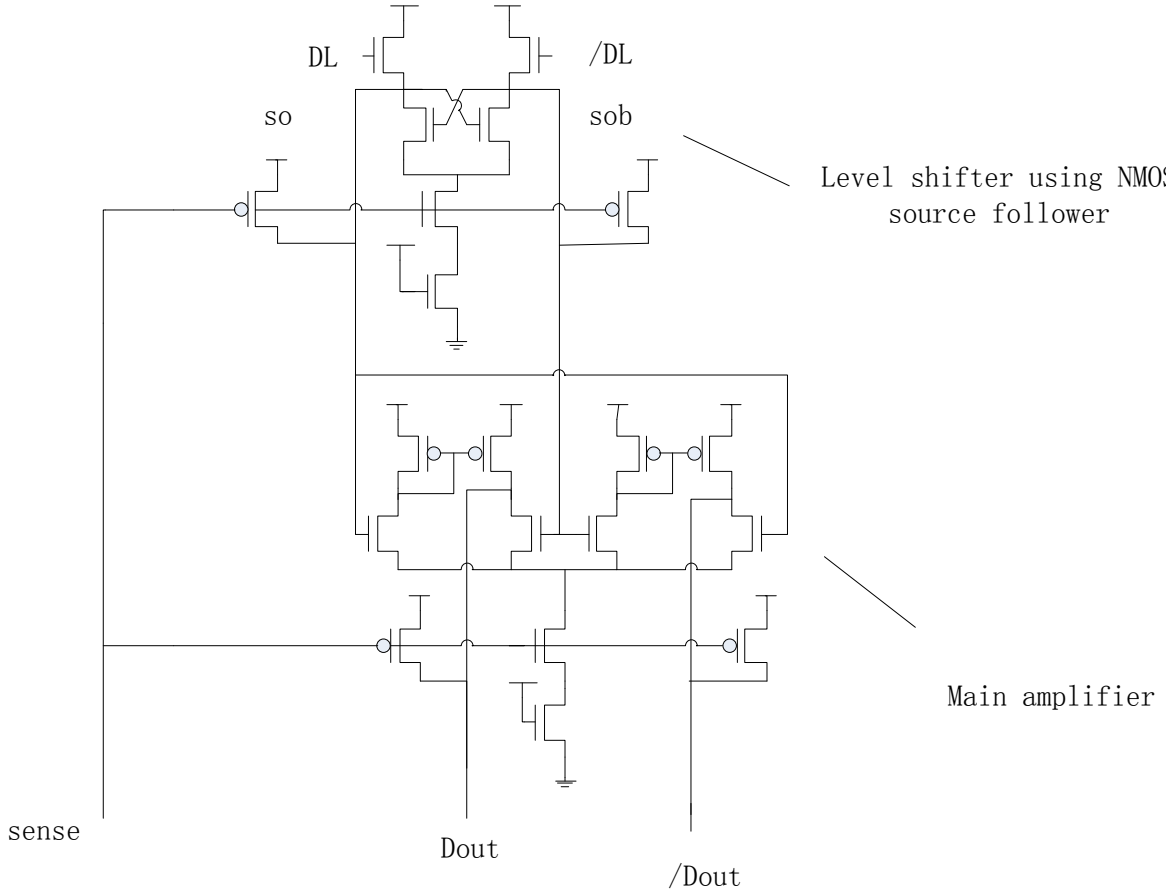
- 尺寸参考下一页图

灵敏放大器


SENSE
vdd
MP2
MP0
MP1
MP3
pmos_3p3
"pmos_3p3"
totalW=2.15e-6
totalW=1.35e-6
fingerW=2.15e-6
fingerW=1.35e-6
l:350e-9
nf:1.0
m:1.0
vdd!
READ_OUT
READ_OUT_B
READ_OUT
SENSE
READ_OUT_B
MN0
MN1
MN2
MN3
MN4
nmos_3p3
"nmos_3p3"
totalW=750e-9
fingerW=750e-9
gnd!
net068
net060
net28
bbar
b
totalW=1.5e-6
fingerW=1.5e-6
gnd

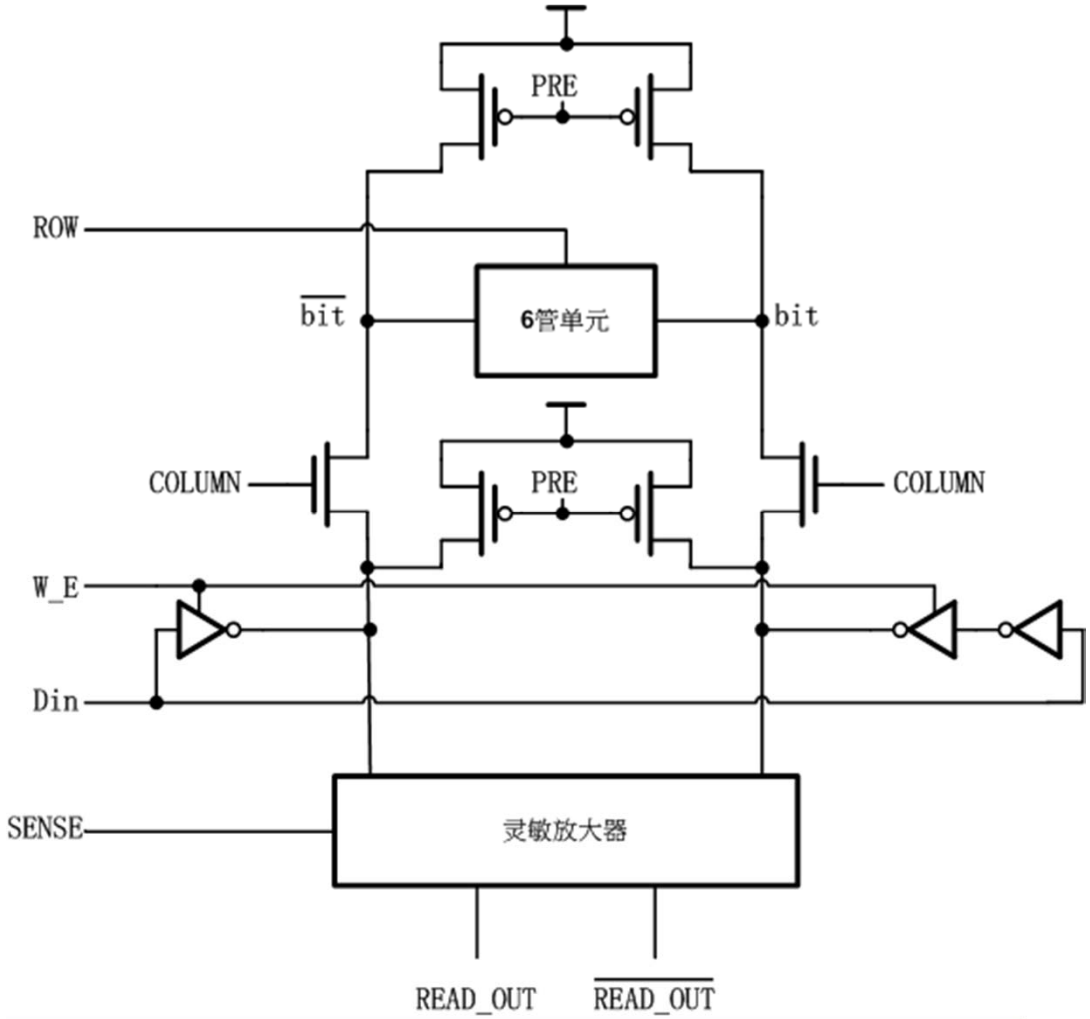
灵敏放大器

- 除了上述锁存式的灵敏放大器之外，另外为大家提供一种灵敏放大器结构供大家参考



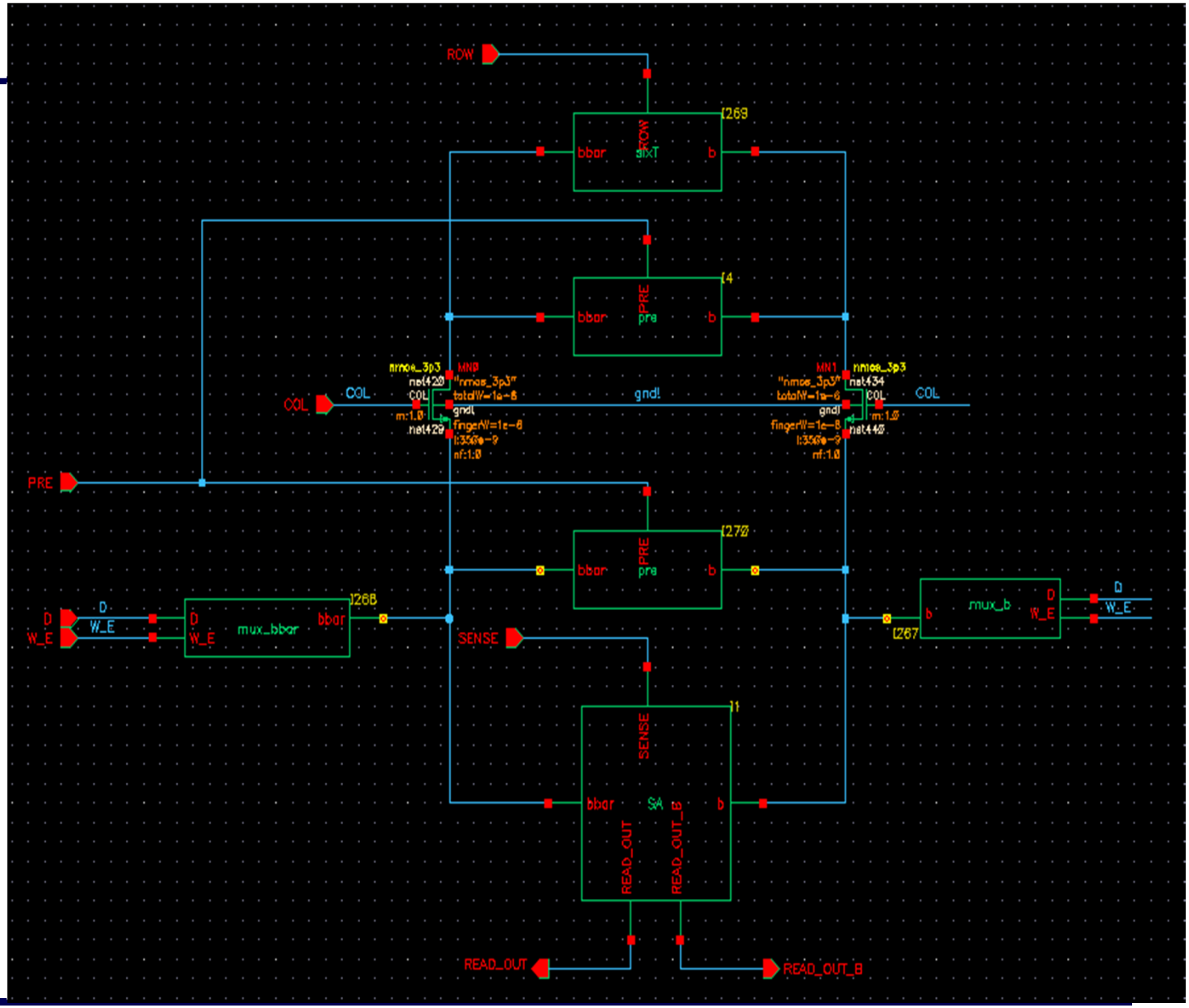
模块连接

- 基本的小模块已经完成，这时候需要把各个模块连接起来
  - 具体的连接关系如图。
  - 列地址控制的NMOS管隔断6管单元和灵敏放大器的位线。
  - 各自的位线都需要有预充结构。



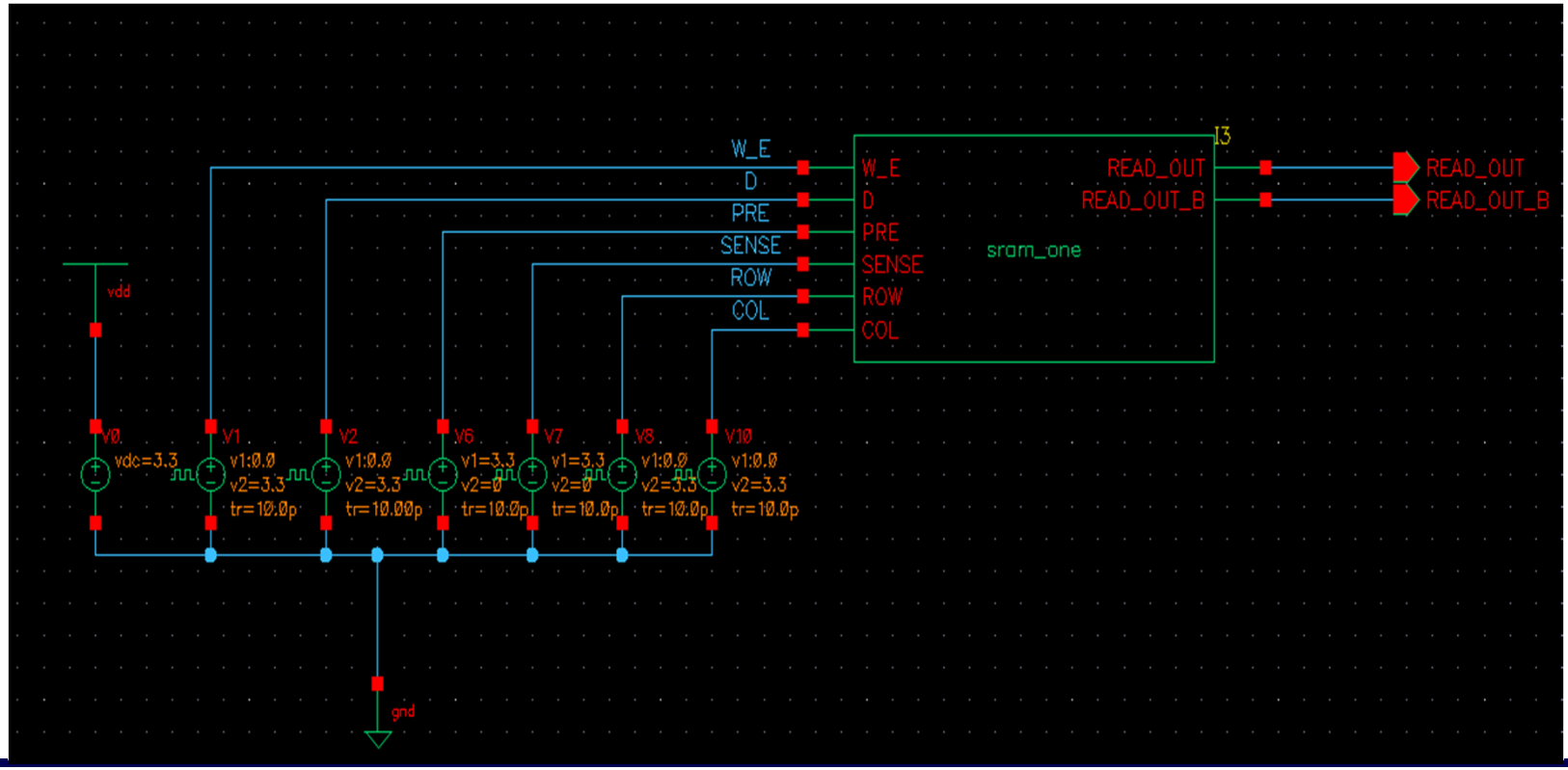
模块连接

- sixT
  - 六管单元
- Pre
  - 预充电路
- SA
  - 灵敏放大器
- mux_bbar
  - 数据输入1
- mux_b
  - 数据输入2



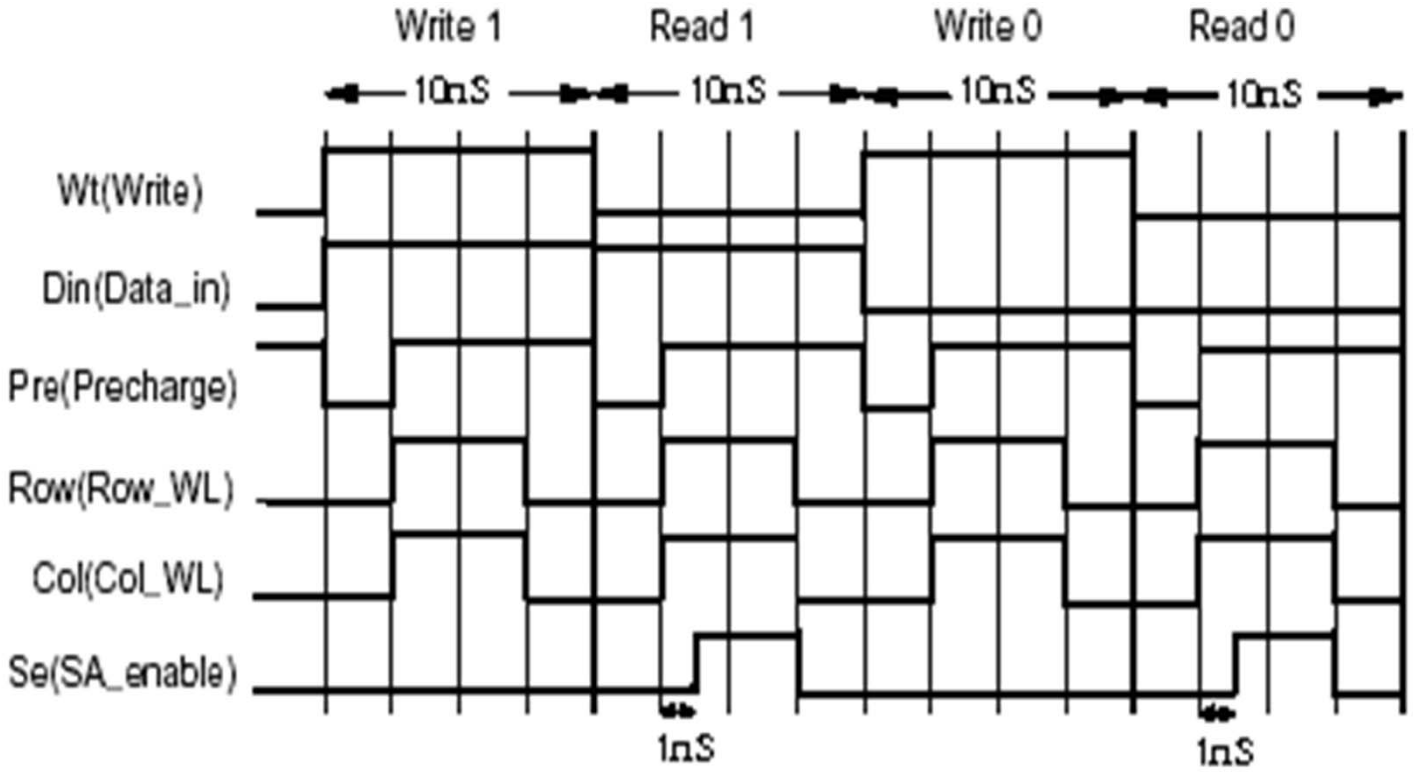
单管仿真

- 再进行一下步之前我们需要保证前面的电路能正常工作。所以我们需要进行一次单管的仿真来验证sram功能的实现。
- 单管仿真只是不涉及地址的变化，没考虑位线电容，其他功能都能够验证。仿真电路如下：



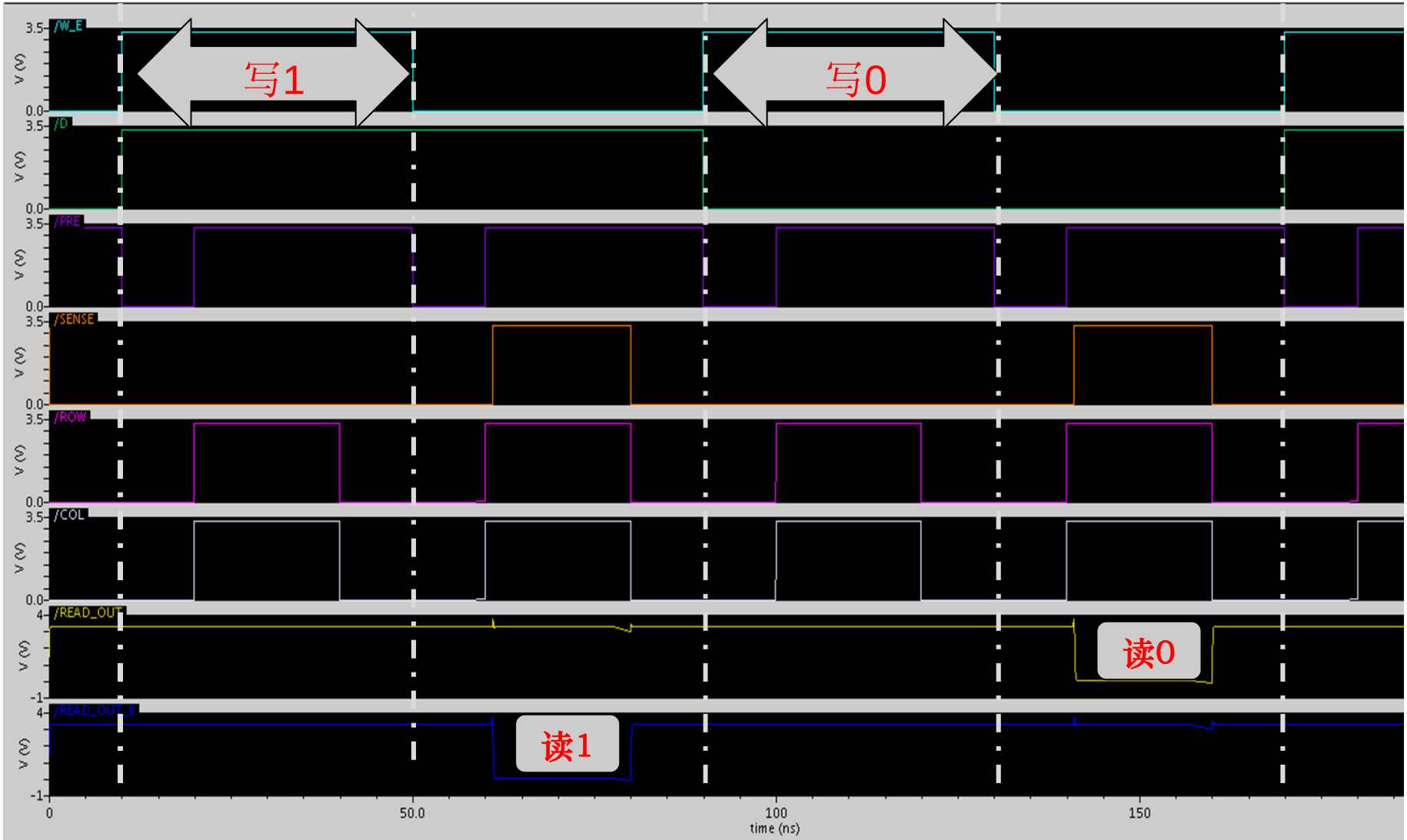
单管仿真参考时序

- 单管仿真的控制信号参考时序如下图所示：



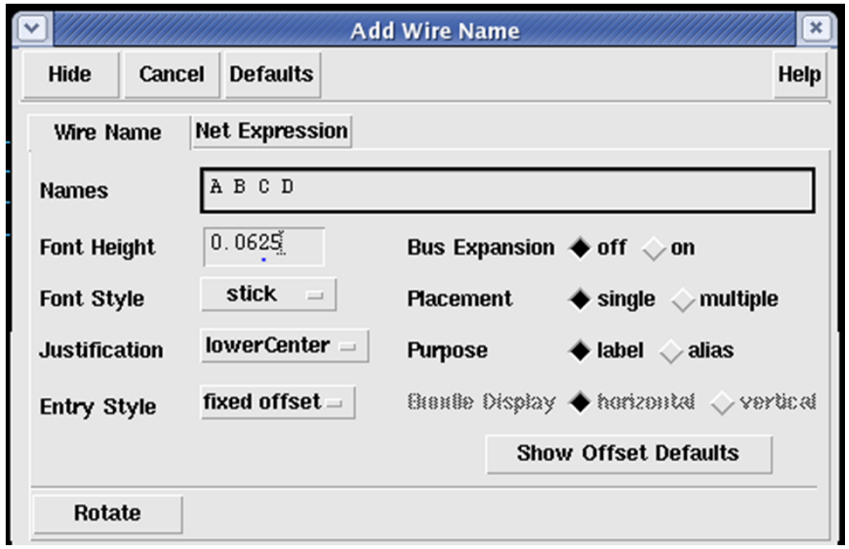
单管仿真

●得到仿真波形如下



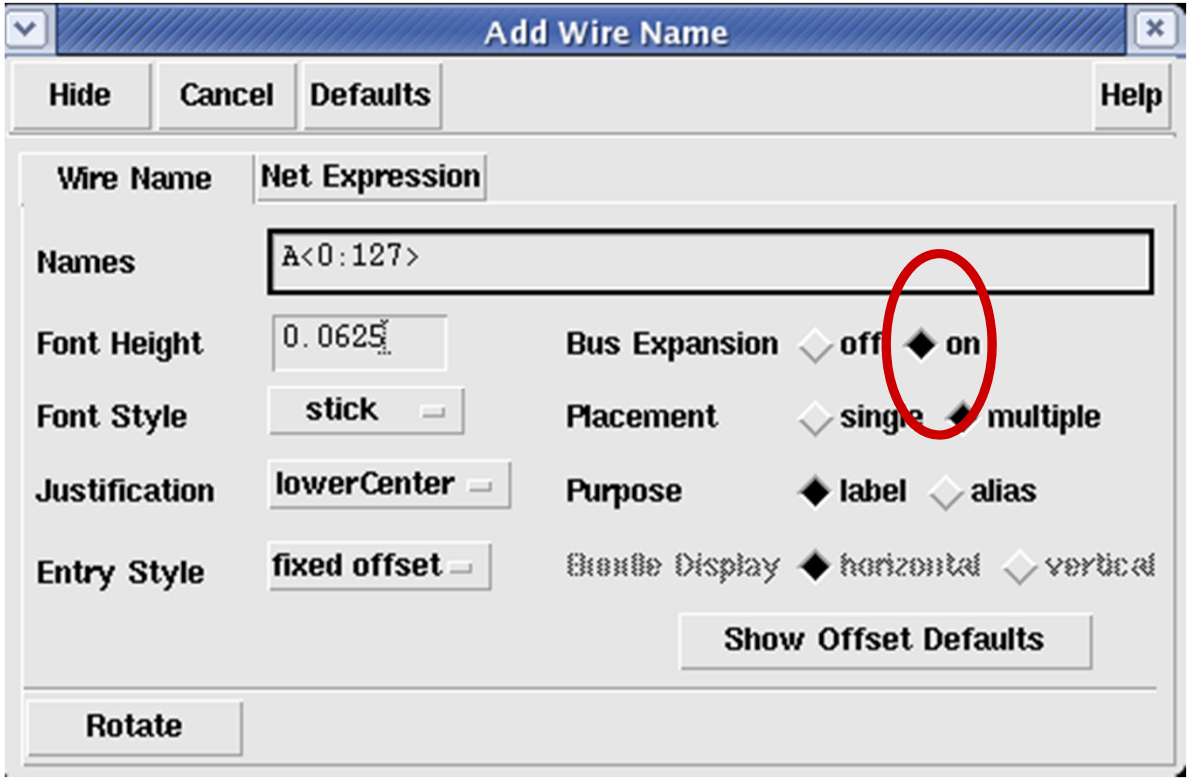
关于Cadence使用

- 在开始设计SRAM阵列和译码器设计之前，有关于cadence使用的一些方法需要说一下。
  - 第一，前面说过了，名字相同的线是连在一起的。另外如果线的名字和某个pin的名字是一样的，也表示二者是连在一起的。
  - 第二，如果需要加多个pin或者label，可以在添加的names一栏填多个名字以空格隔开。如图：



关于Cadence使用

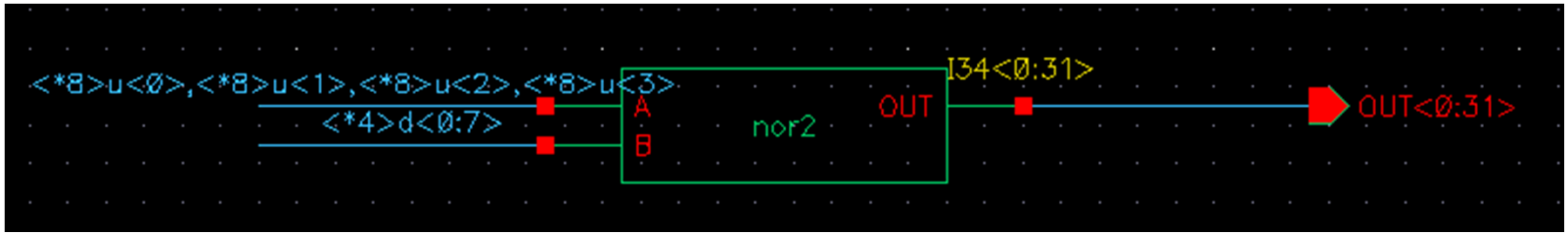
- 第三，举例说明：如果需要将128根连线分别命名为A<0>,A<1>,……,A<127>。则如图（如果不选中multiple，则需要一个一个的添加。如果不选中on，则会将此根线定义为总线，相当于128根线，具体用法下面会讲）。



关于Cadence使用

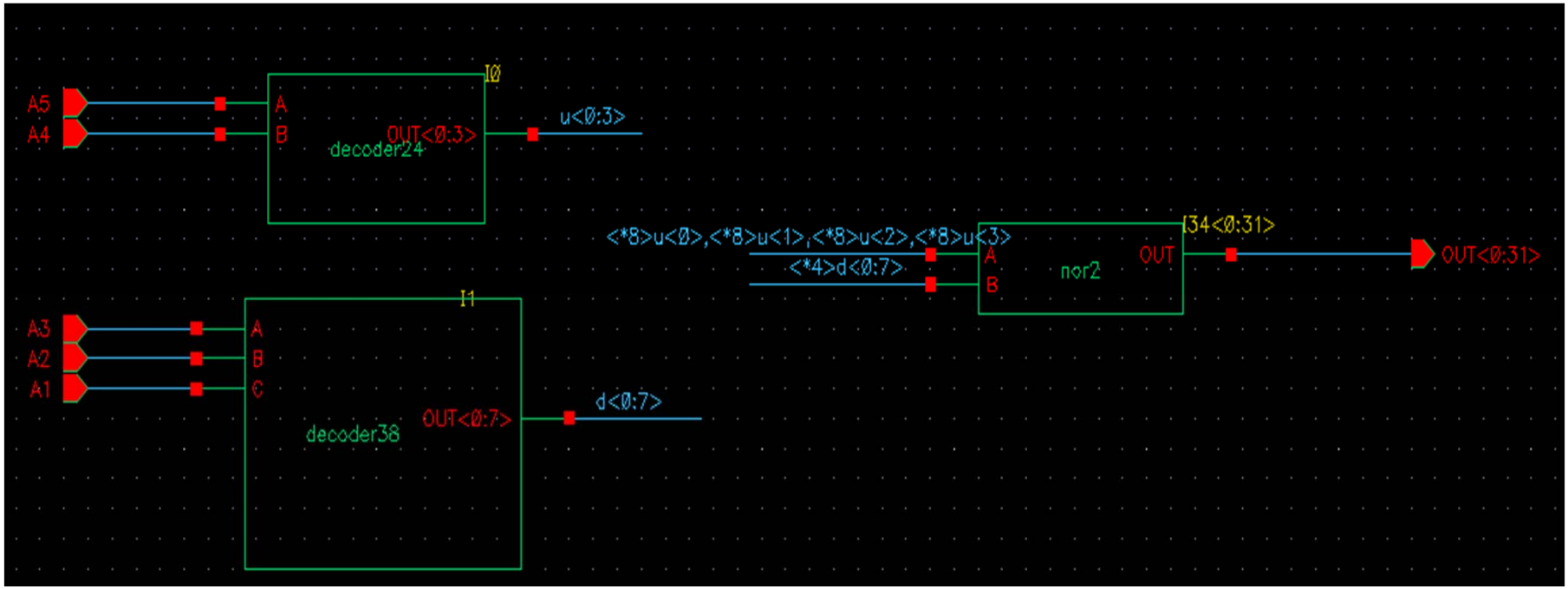
- 第四，总线使用。举一个例子：图中我添加了32个2输入或非门（添加的时候names一栏填I<0:31>, 注意I是向量名字，要和其他的不同，这里是“I34”。<0:31>表示有32的器件名字分别为I<0>,I<1>,……,I<31>）
  - 输入B的wire name为<*4>d<0:7>，<*4>表示循环4次。就是说B0~B7连接d<0>到d<7>,B8~B15连接d<0>到d<7>,B16~B23连接d<0>到d<7>,B24~B31连接到d<0>到d<7>。
  - 输入A的wire name为<*8>u<0>,<*8>u<1>,<*8>u<2>,<*8>u<3>，同样<*8>表示8次。注意中间用逗号隔开了。表示A0~A7都连u<0>，A8~A15都连u<1>,A16~A23都连u<3>，A24~A31都连u<3>。
  - 输出OUT总共32根线，直接命名为OUT<0:31>

注意这里加wire name的时候不用选择bus expansion的on



译码器

- 行译码器
  - 行译码器是一个5-32的译码器。可以自己选择实现方式。这里示例是用一个2-4和一个3-8译码器构成了行译码器（前一级的2-4和3-8都是输出的非信号，所以后面用或非门产生最终输出）：



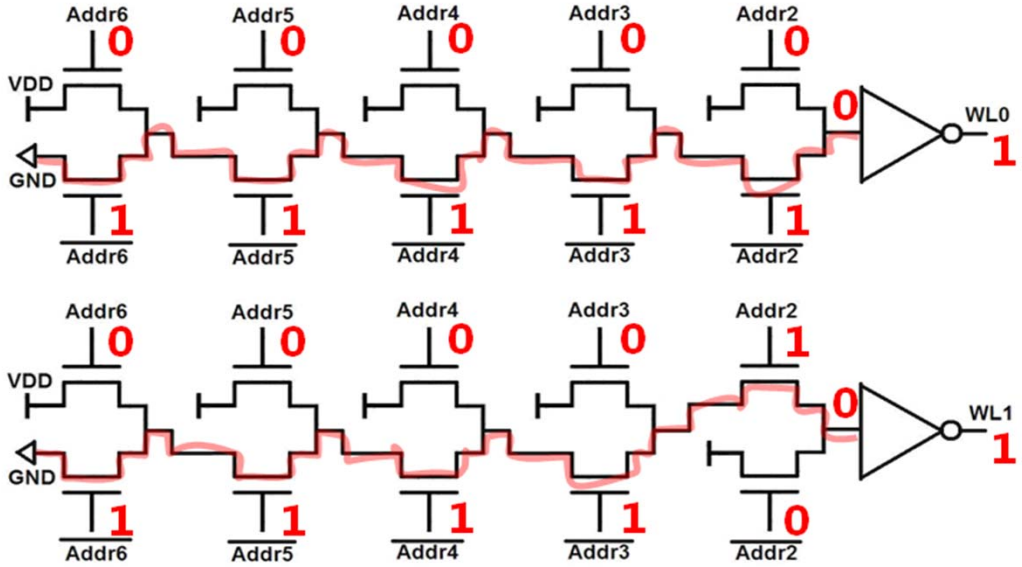
  - 行译码器设计完后建议大家仿真一次，验证其功能是否正确。

译码器

- 行译码器
  - 行译码器也可以直接用32个5输入与门构成
  - 单元库中并没有5输入与门，需要自己用单元库中的单元或是直接用MOS管搭建
  - 这种方法最大的缺点就是连线复杂，版图的面积也比较大

译码器

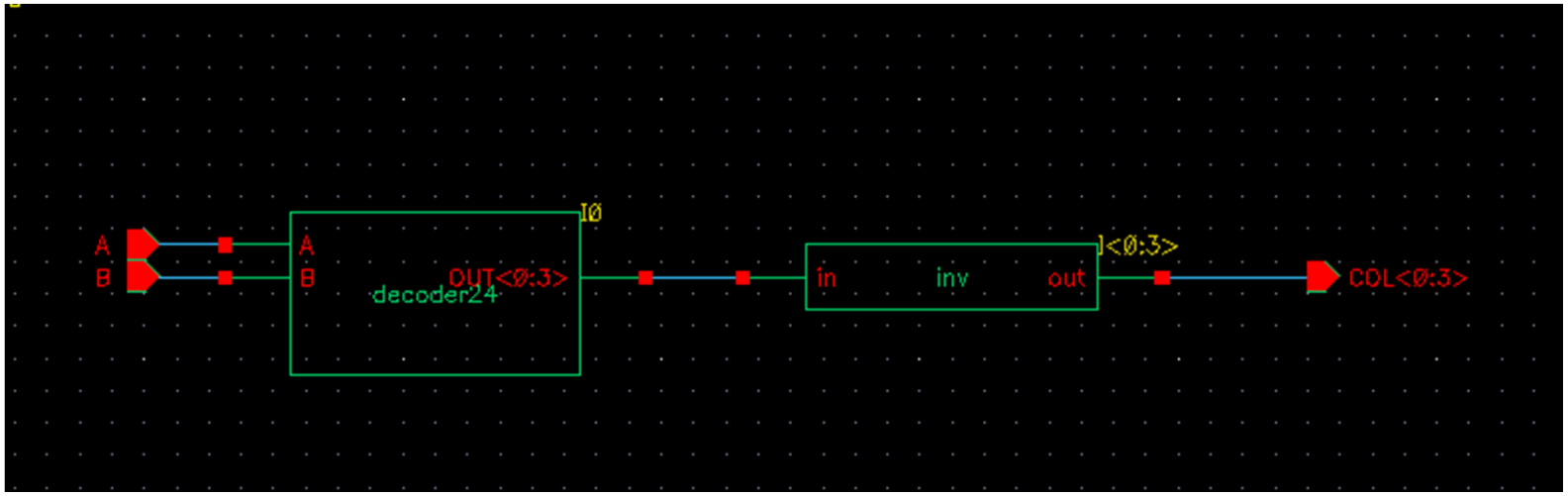
- 行译码器
  - 还可以采用一种Pass-Transistor结构的译码器，既可以节省面积又方便布线，如下图所示。
  - 当且仅当地址信号为00000时，低电平能被传输到WL0译码电路反相器的输入端，从而使输出为高电平，即选中该字线；同理，当且仅当地址信号为00001时，WL1字线被选中。



译码器

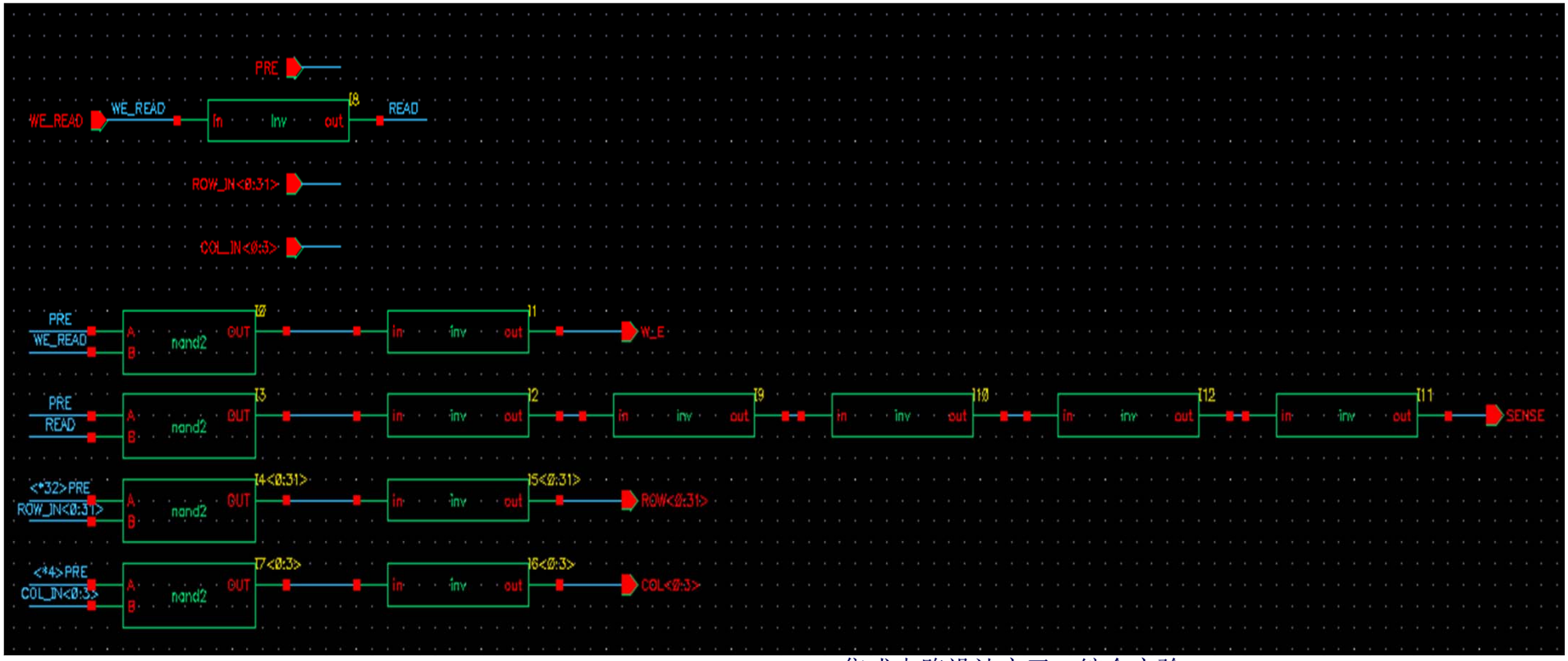
- 列译码器

  - 列译码器就是一个2-4译码器



控制信号产生电路

- 这里主要是为了保证预充期间，读写无效，地址无效。所以各个信号都和预充控制信号PRE进行了与。
- 需要注意的是，SENSE信号（读出控制）必须要在地址信号有效之后才能有效。所以加了延迟缓冲。



SRAM阵列

- 单管仿真正确后，我们可以开始构建SRAM的阵列。整个SRAM是32行32列。行译码输出选择其中的一行。列译码输出选择其中的8列。

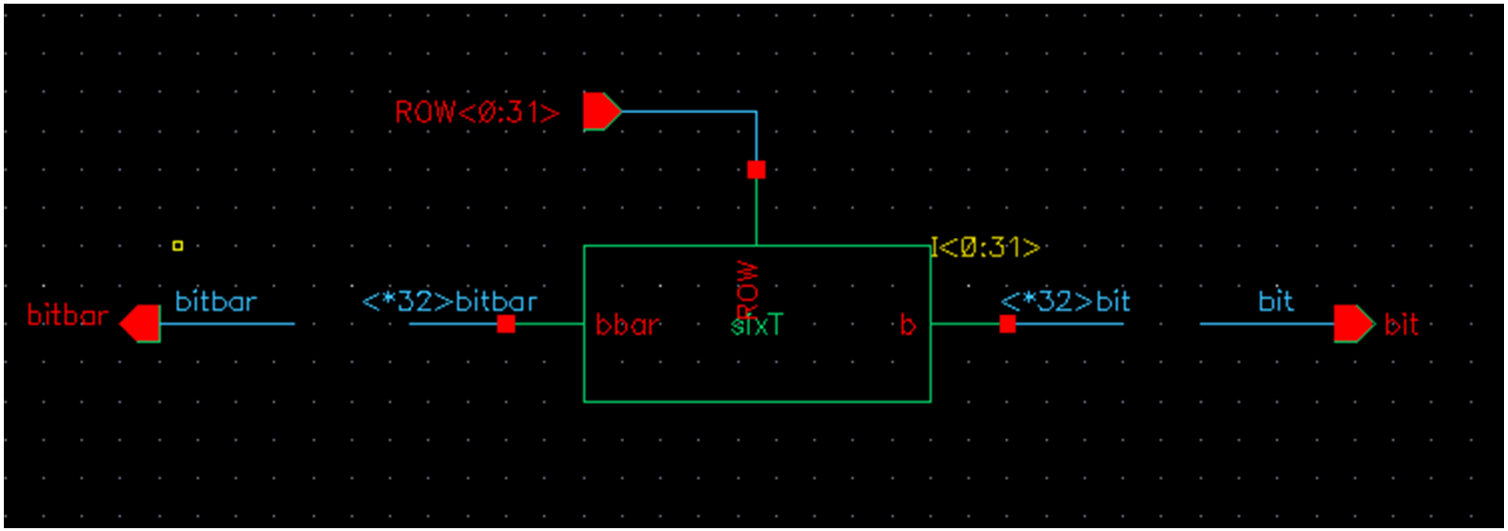
  - 首先，画出SRAM的一列。每一列共用预充电路，数据输入电路以及灵敏放大器。画出一列之后，可以进行仿真验证加入行译码后，电路是否能够正常工作。

  - 其次，由于每8列共用列译码输出，所以可以接着构建出8列作为一个模块（当然也可以跳过这一步，直接构建32列）。

  - 最后，完成32*32的SRAM阵列。并进行仿真验证。功能正确后可以开始版图设计。

SRAM阵列 --- 一列

- 一列6管单元总共32个。如图，他们的位线都是连在一起的。但是行地址是分开的。

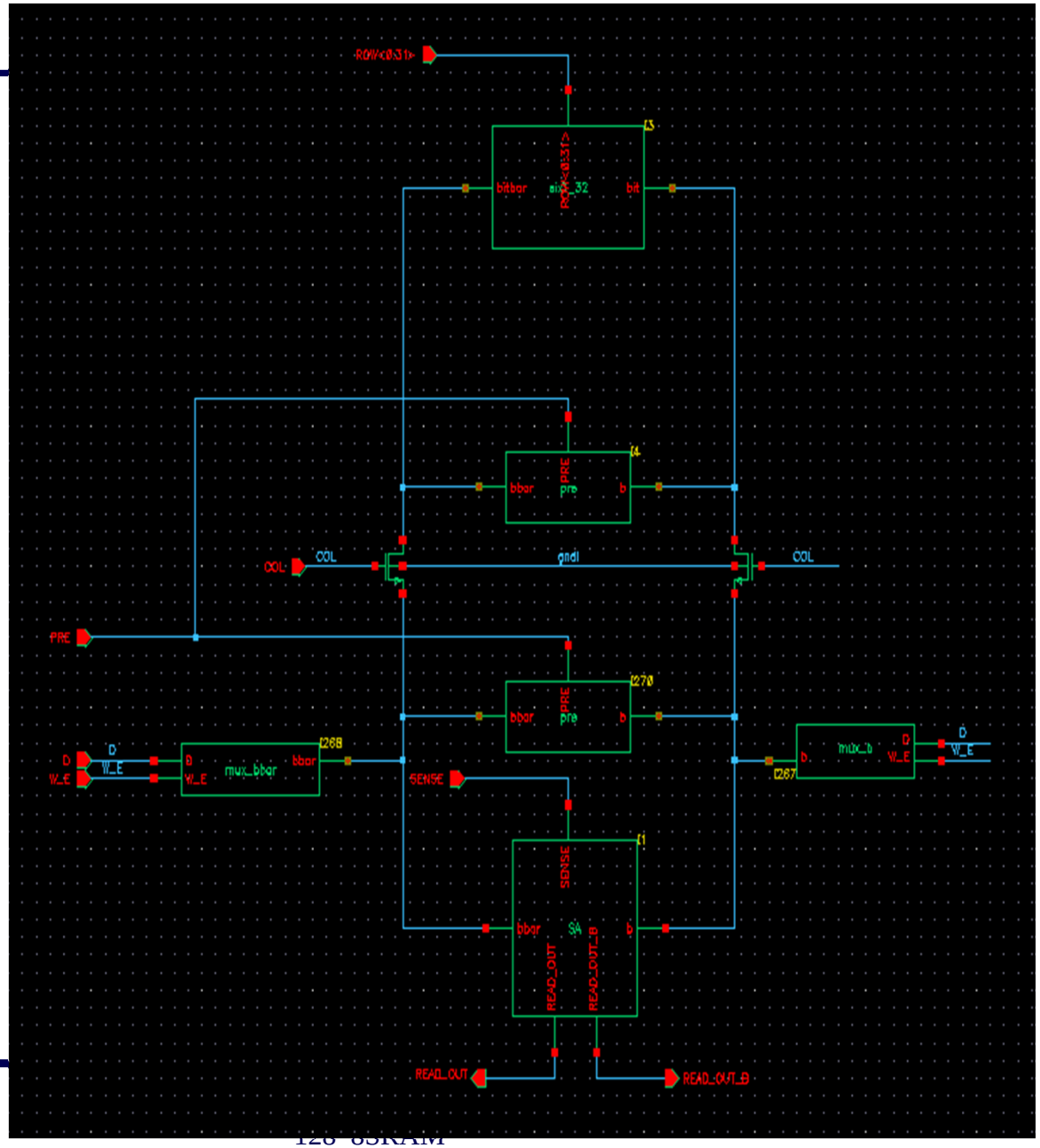


SRAM阵列 --- 一列

●SRAM的一列除了32个共用位线的6管单元，还有一套预充电路，数据输入电路和灵敏放大器。

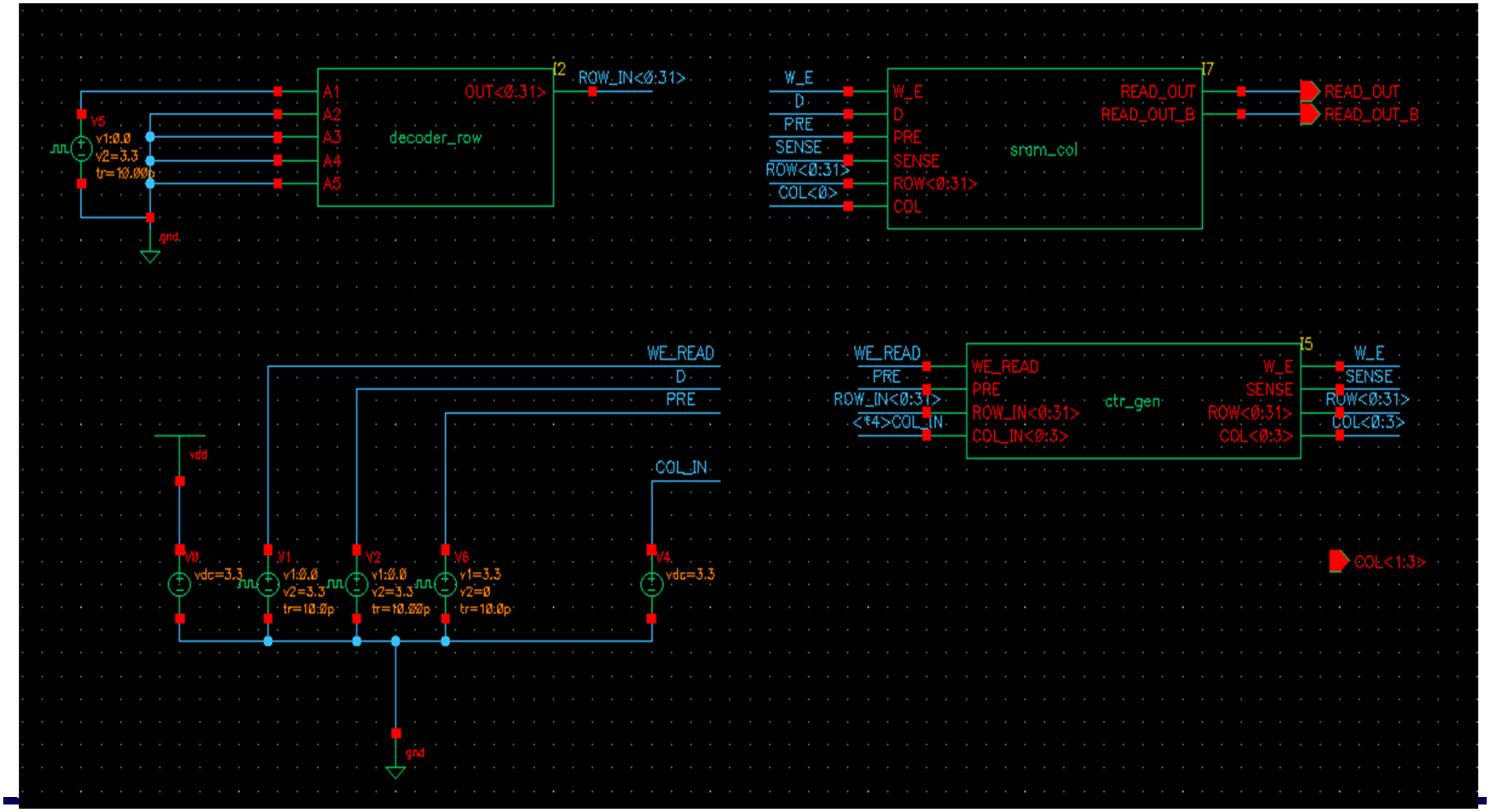
●连接方式和之前一样，只是把单个6管换成了32个6管单元。

●一列画完后，建议大家进行仿真验证功能。



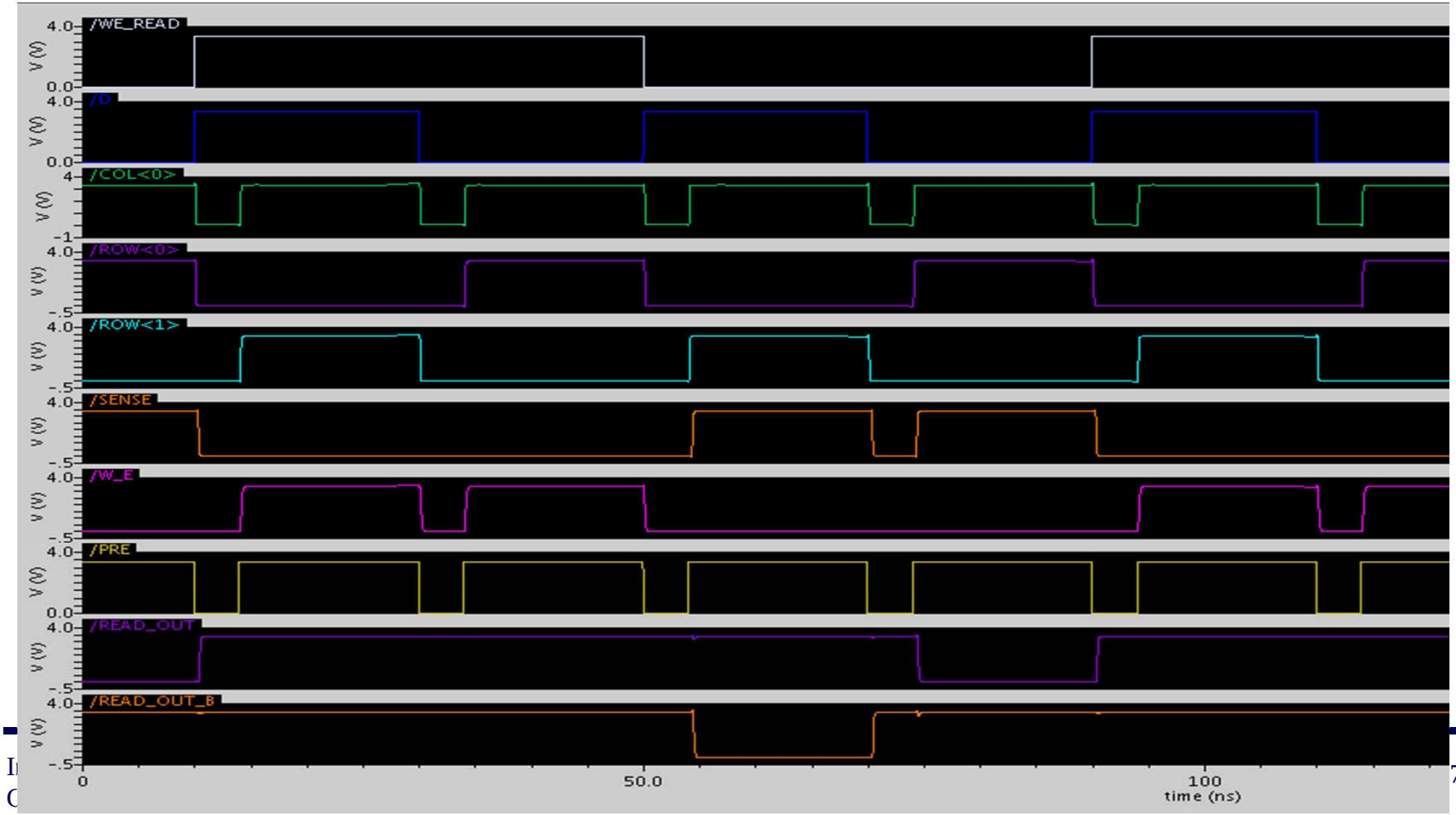
SRAM阵列 --- 一列的仿真

- 结合前面的行译码器以及控制信号产生电路进行仿真。列地址设为总是有效。



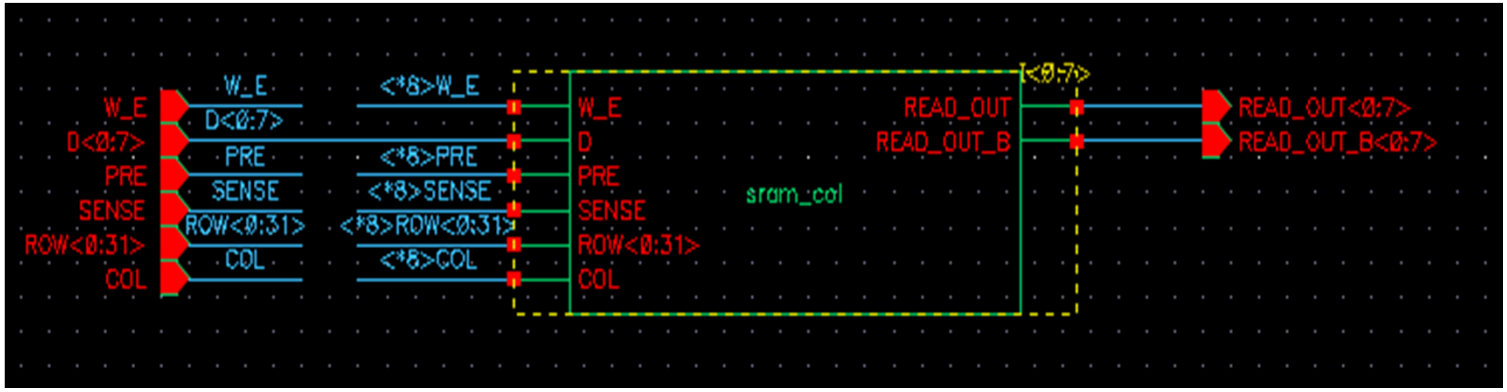
SRAM阵列 --- 一列的仿真

- 仿真波形如图。这里观察了内部产生的控制信号。建议大家另外观察一下位线信号的变化以及存储单元内部信号变化，以便更清楚的理解SRAM的工作过程。

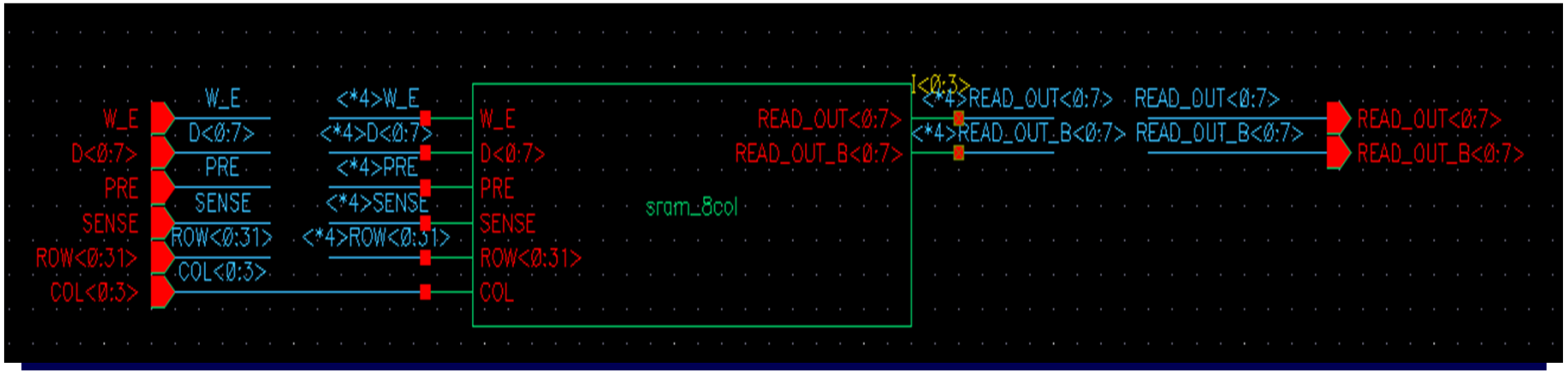


SRAM阵列 --- 32列

- 这里将8列作为一个模块。如图调用了8个sram_col的模块。连接关系通过label来标明。



- 然后再将4个8列的模块构成32列的阵列。

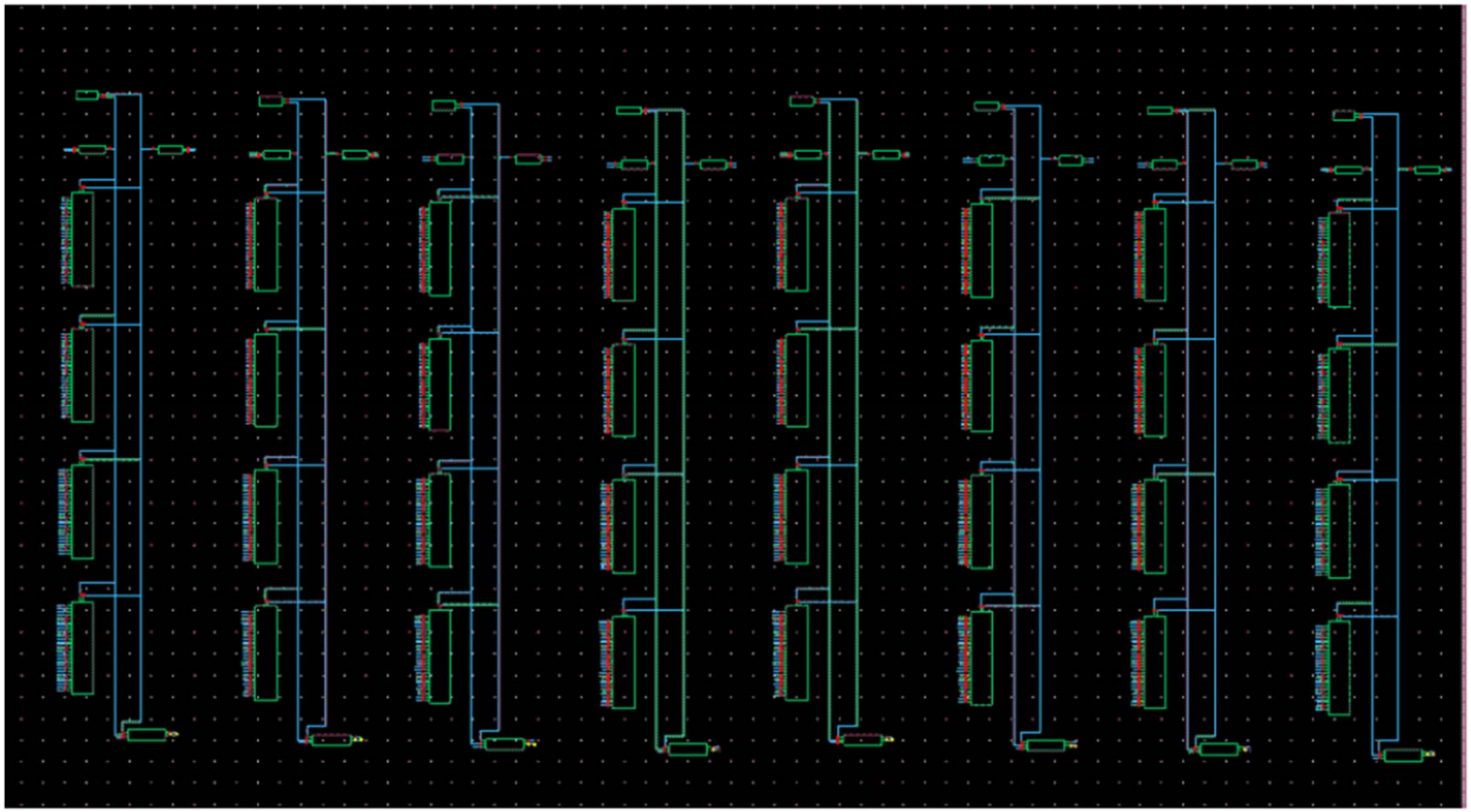


SRAM阵列 --- 32列

- 同样可以将4列作为一个模块
- 这4列6管单元可以共用一个灵敏放大器和数据线
- 再将8个4列模块组成32列单元阵列
- 进行读写操作时，每个4列模块只是读写1bit的数据，8个4列模块完成1字节数据的读写

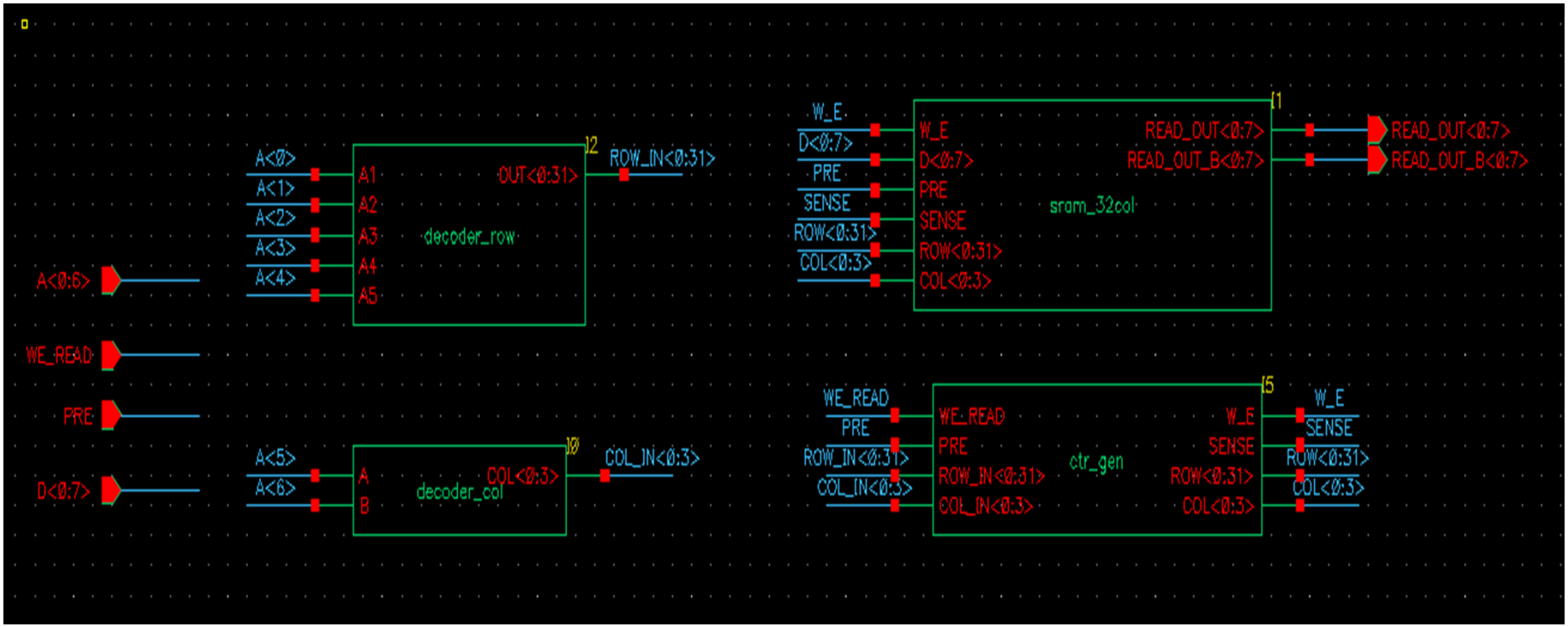
SRAM阵列 --- 32列

- 当然，用总线的方式结构没那么清晰，那也可以用一列一列都画出来。示意图如下：



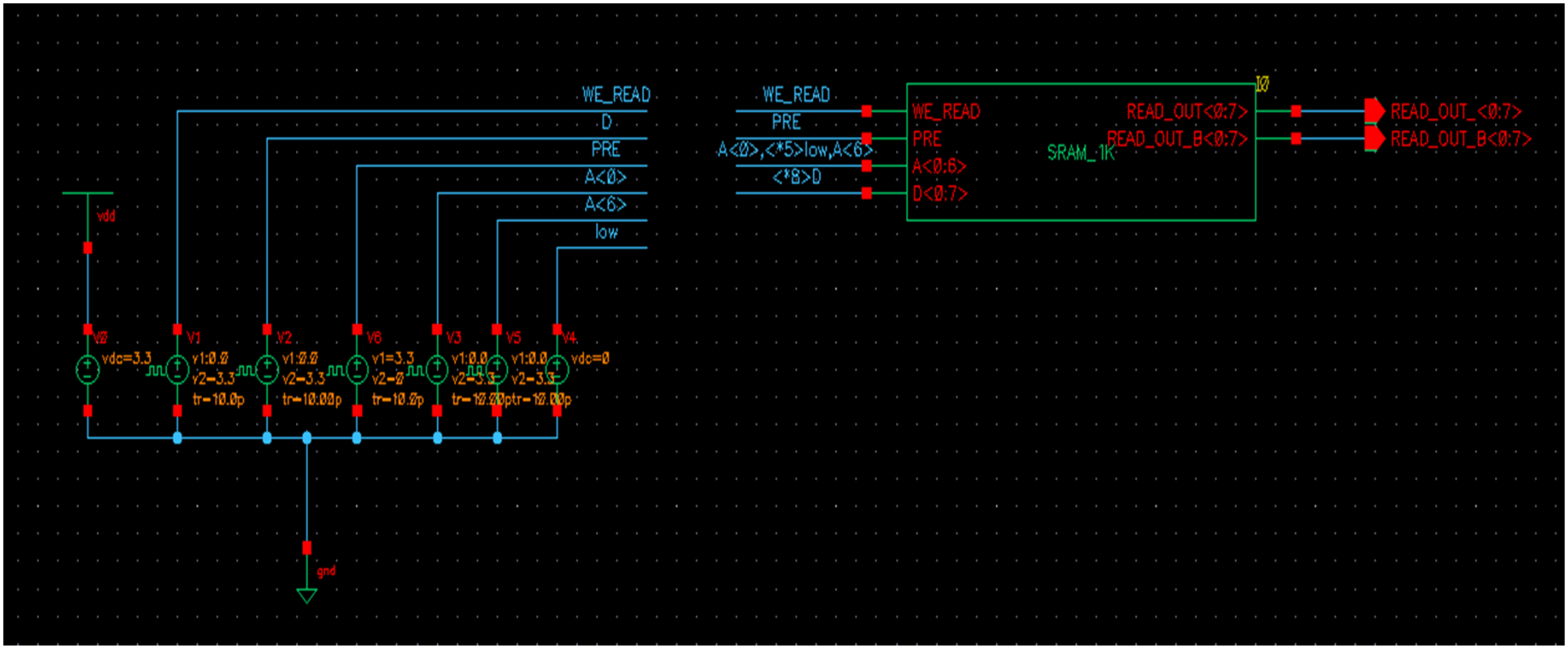
1028bit SRAM

●所需要的各个模块都已经设计完毕，现在需要将各模块连接起来。



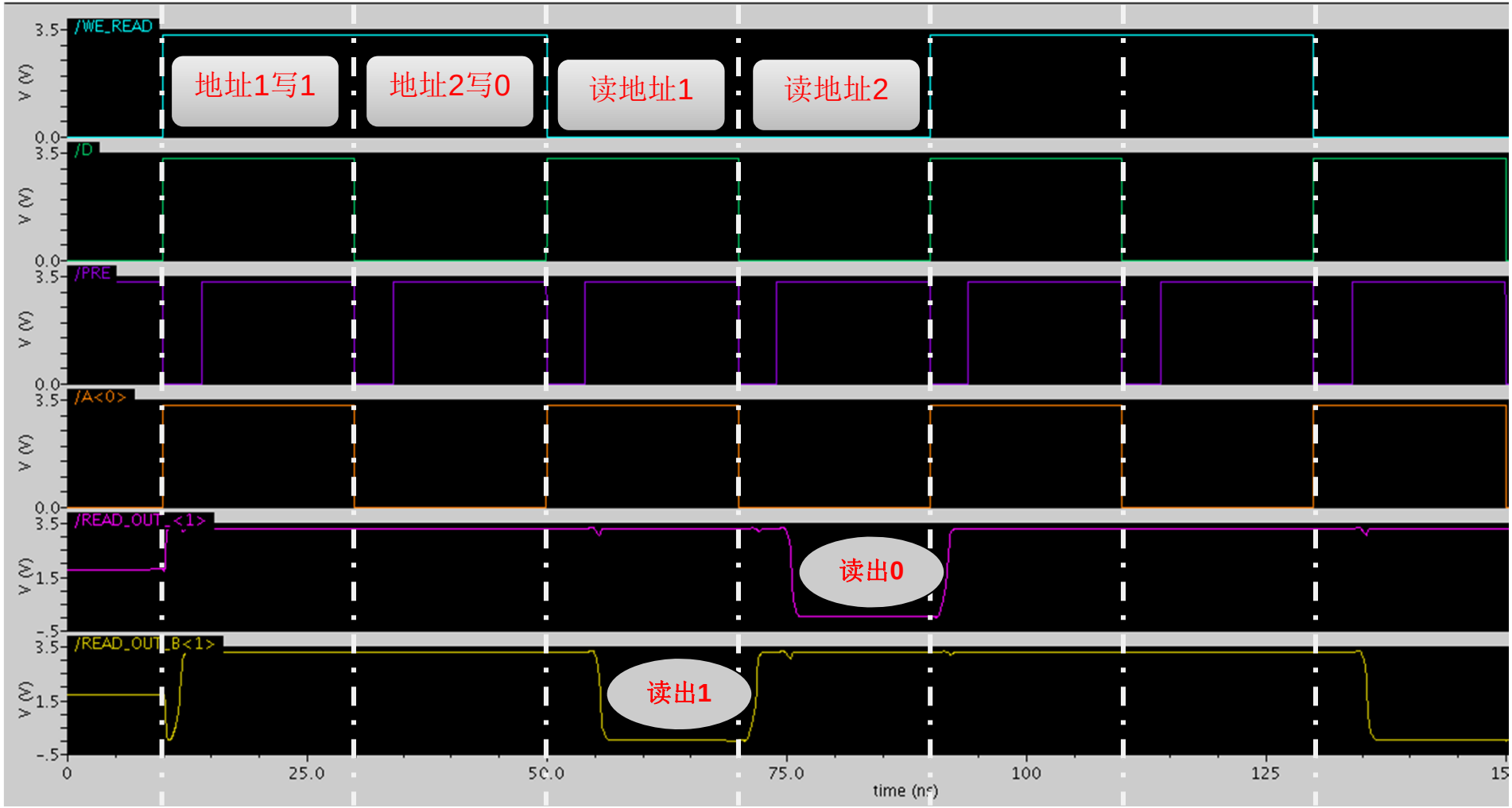
仿真验证

- 电路画完就可以进行仿真验证了。



仿真验证

- 仿真波形如图。

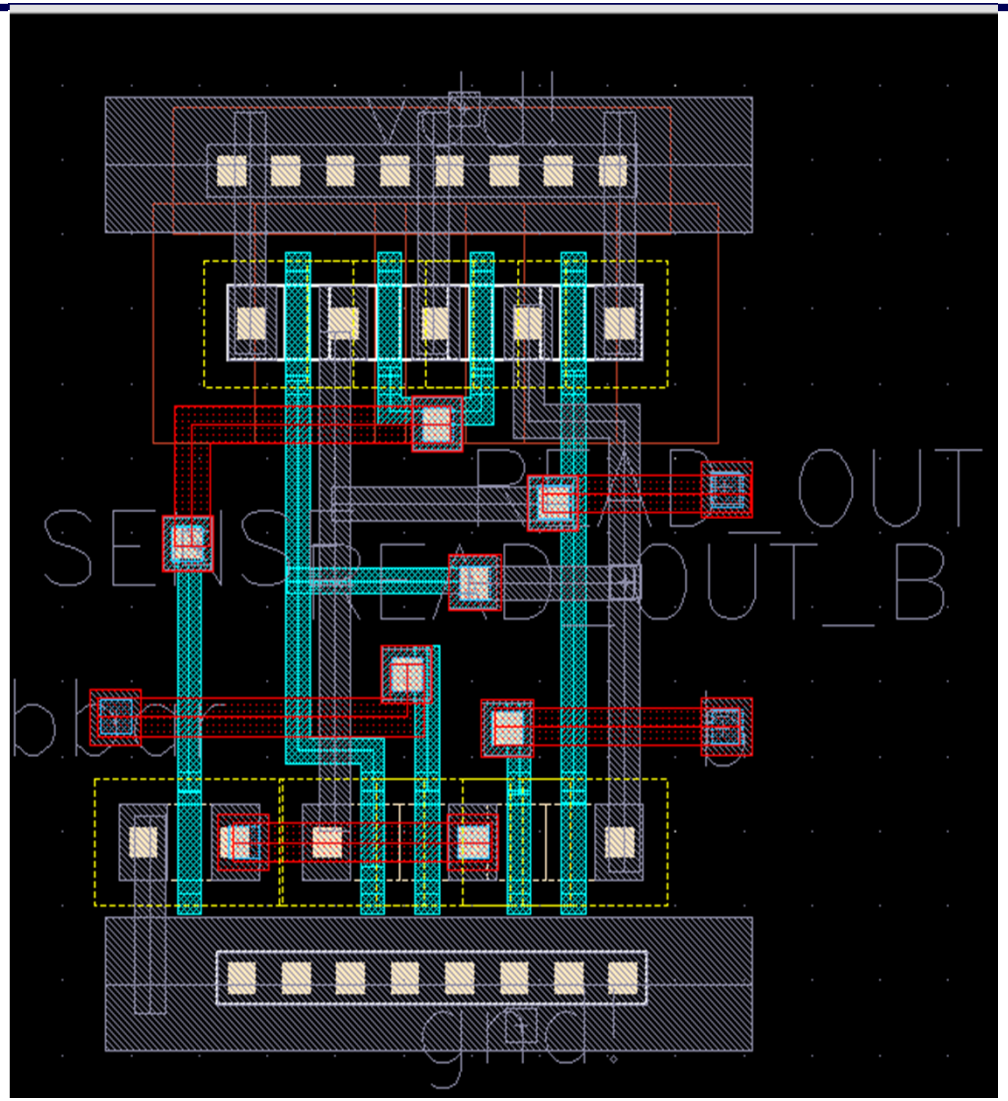


版图设计

- 电路验证完毕，可以开始进行版图设计。
- 在版图设计的时候，需要注意下面几个问题：
  - 第一：MOS管的衬底不能空接。记得添加M1_NWELL和M1_PUSB。
  - 第二：添加vdd和gnd金属线的时候，因为需要走的电流比较大，所以建议宽度都取1.8um(与标准单元库一致)。
  - 第三：画小模块版图的时候，最好是将版图放到第一象限，贴近零点处。
  - 第四：所有模块的vdd以及gnd的线记得要连一起。
  - 第五：译码器和6管阵列的高度设计好，尽量一致，使得版图规整。

版图设计

- 版图设计也跟电路设计一样，首先从小的模块电路开始，一步一步的往下做。
- 每画完一个小模块电路的版图都要DRC和LVS正确，才能进行下一步。
- 版图设计时，布局自由度比较大，大家自己考虑设计。
- 图中给出的灵敏放大器的版图，供大家参考。
- 另外，设计的时候建议大家将6管单元版图尽量缩小，因为6管单元占主要的面积，其他部分对面积影响不太大。

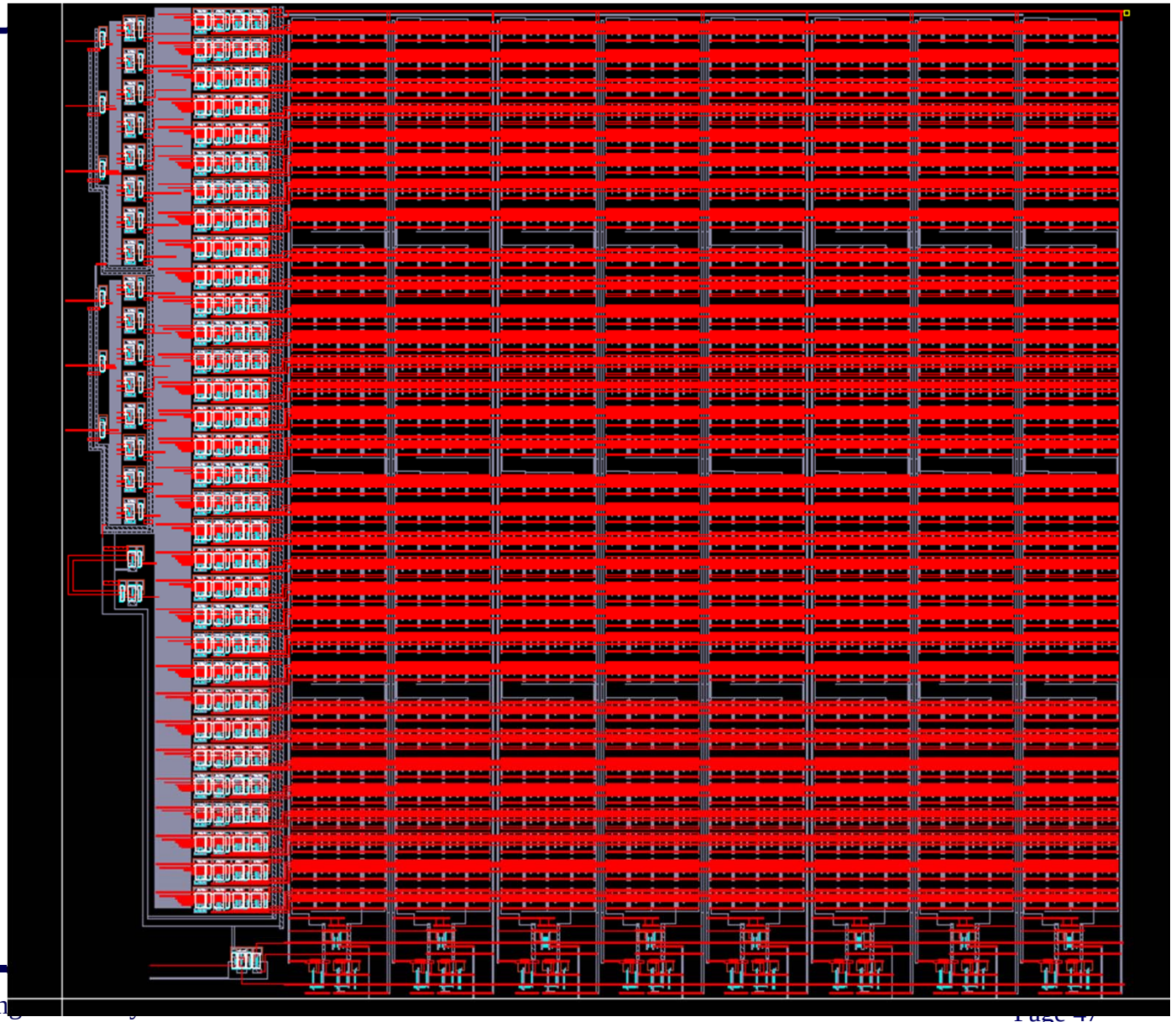


版图设计

- 版图设计时，经常需要查错（包括DRC和LVS的错）。
- DRC查错
  - 做完DRC后报错，可以：verify->markers->find进行差错。在弹出框中选择zoom to markers，点apply或者next就可以将屏幕移到出错处。
  - 当然在icfb的CIW窗口也会列出出错的数量和原因。
- LVS查错
  - LVS完之后可以在output中查看是否match，哪些地方不match，以及不match的原因。
  - 另外如果想定位到具体位置，可以点击Error Display，会弹出一个窗口。注意，如果想定位到schematic中，那就先点击schematic的窗口，再点弹出窗口中的OK。如果想定位到版图中，需要先点击extracted的窗口，再点弹出窗中的OK。

整体结构

- 将各个小模块完成之后，组合成最终的版图。
- 右边给出了一个整体的参考结构。注意这个版图和前面的电路结构不是完全一样的，主要是给大家参考一下整体结构的布局。



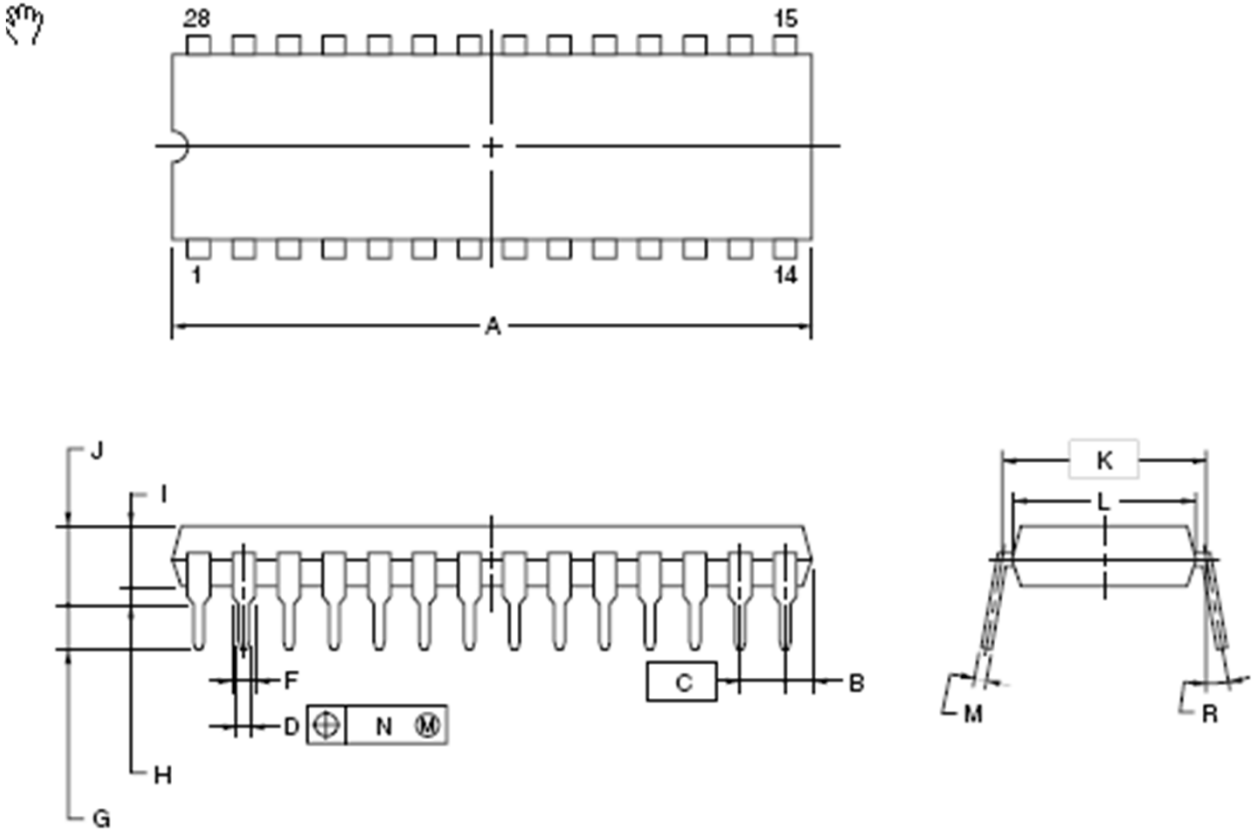
仿真和验证

- 版图画好之后，就可以进行后仿真。

- 测试电路功能和性能，和前仿真进行比较。

封装形式

- 双列直插陶瓷封装
- 16，18，22，24，28等封装



测试要求

- Agilent93000SoC测试仪
- DIP40_A标准测试板

参考资料

- 甘老师“CMOS”书中存储器和IO部分
- Rabaey的书

测试安排

- 本项目的流片时间初步定在12年1月，流片完成时间是12年4月
- 封装后，会安排参与流片的项目组进行测试，时间应该在下个学期

实验验收

- 学生成绩由3部分组成
  - 1、课上完成情况及质量（最后一次课验收）。
  - 2、答辩（制作ppt文稿对自己的实验进行陈述并回答老师的提问）
  - 3、实验报告（答辩完之后提交）
- 讲义中的结构只是供大家参考，鼓励大家在实验中加入自己的创新。有创新的同学将予以适当的加分。