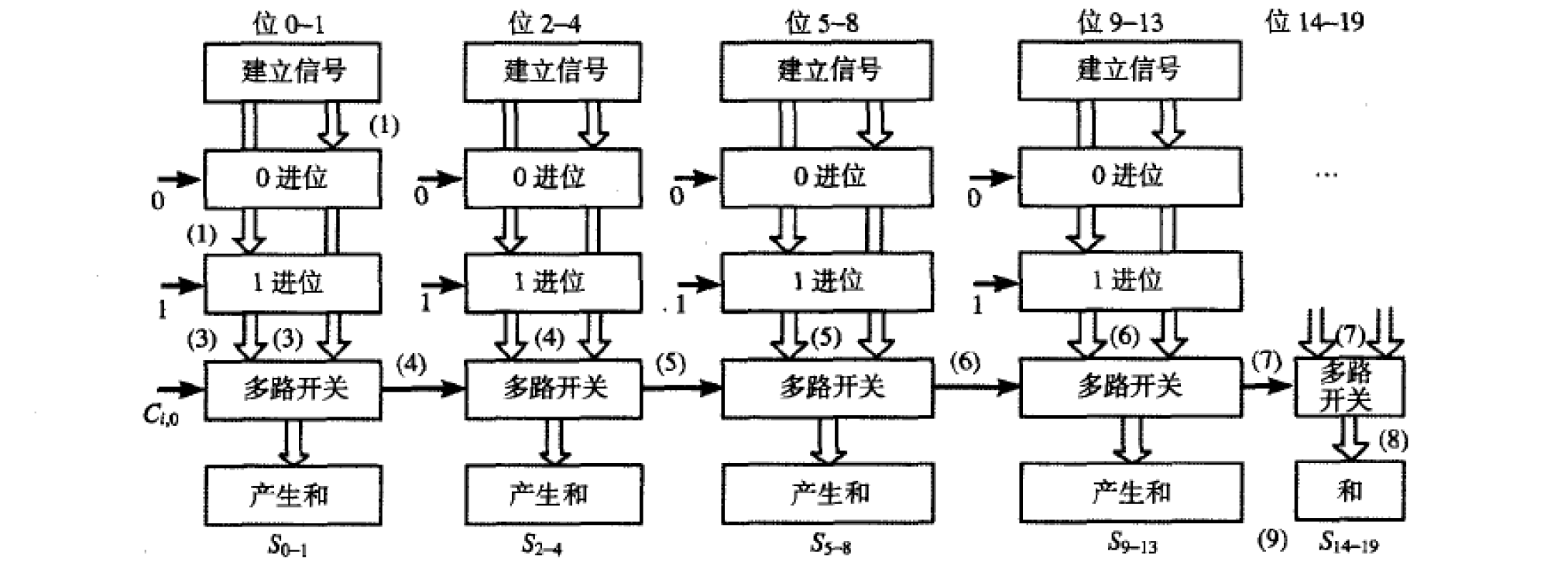
**数字集成电路大作业报告**

樊子辰 无54 2015011065

**一 结构设计**

在32位加法器设计的过程中，我们采用了多种结构并进行比较。对于进位的处理，我们采用了平方根选择进位加法器。平方根选择进位加法器与行波进位和线性选择进位的结构相比速度较快，而与超前进位结构相比功耗较小，于是平方根选择进位结构便成为了我们的不二之选。



平方根选择进位结构

对于1bit的全加器电路结构，我们尝试了两种结构，一种是曼切斯特加法器电路，一种是镜像加法器电路。其中曼切斯特加法器的进位链又有动态和静态两种实现方式。经过比较，排除与仿真，我们最终确定了自己最终的加法器结构——基于镜像全加器的平方根选择进位加法器。

**二 电路设计**

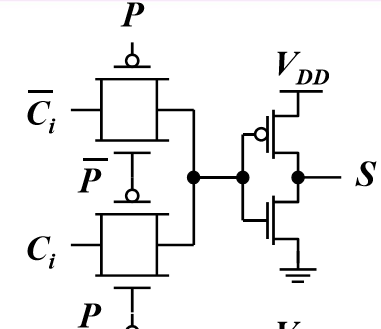
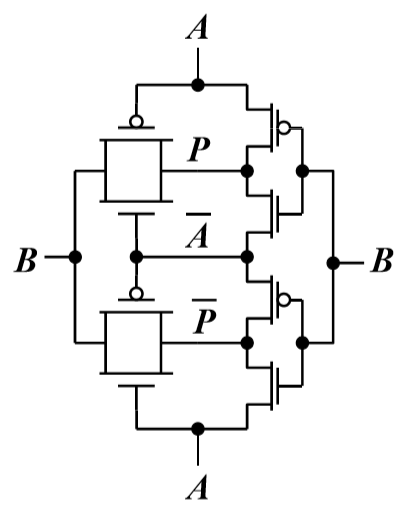
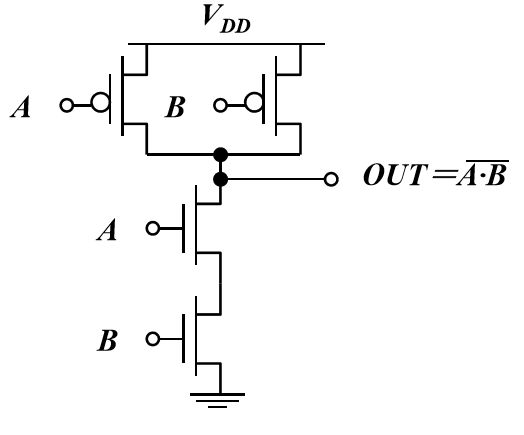
**2.0 全加器逻辑**

为了化简电路结构，我们定义某些中间信号

于是可以将S和Co重新写为P和G（或D）的函数

**2.1 曼切斯特全加器**

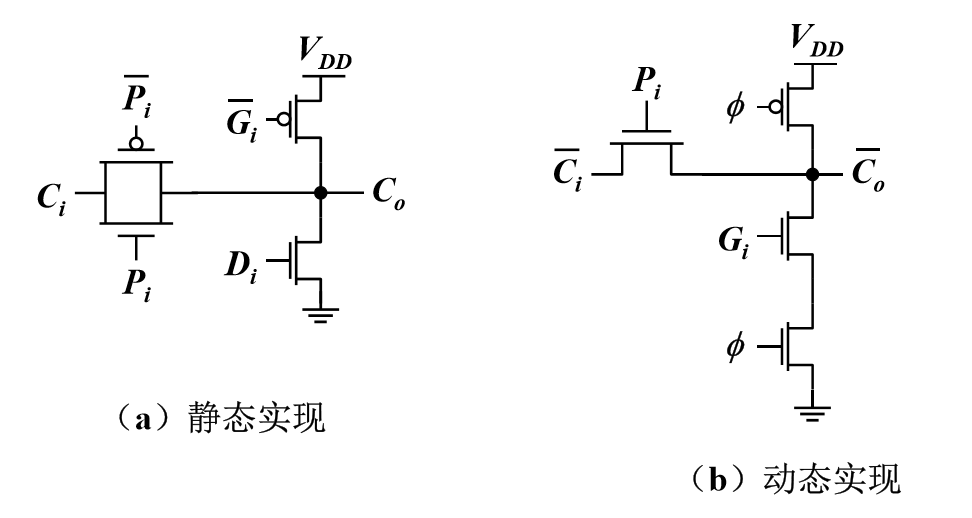
和产生电路与传输门型加法器类似：



用互补静态CMOS 产生P和P‘ 采用传输门的方式来

产生G/G’,D/D’ 产生和位S信号

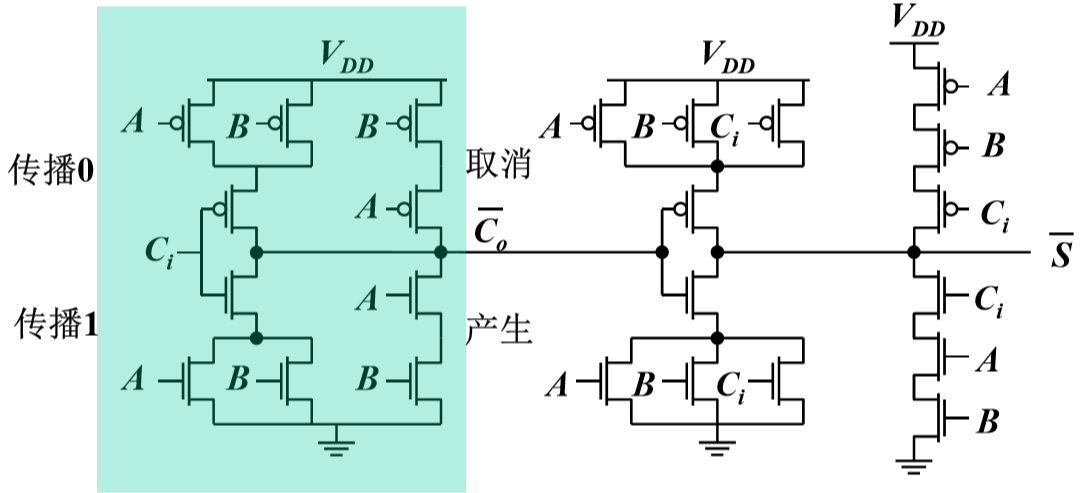
进位链采用曼切斯特进位门的形式：



对于动态实现，我们经过尝试后发现，因为需要设定一个高频clock信号，大量验证加法器正确性会比较困难，于是我们采用了静态实现的方式。

**2.2 镜像全加器**

另外一种全加器结构是镜像全加器，其结构如下：



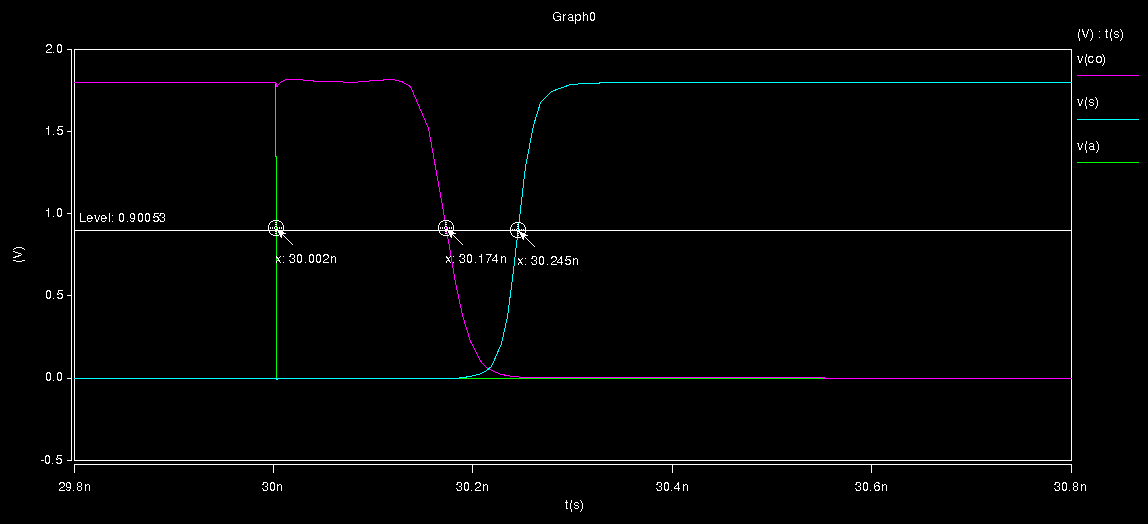
镜像全加器结构

镜像全加器的优点是晶体管数目少、不需要单独的单元产生GDP信号、结构对称且在求和电路中的所有晶体管都可以采用最小尺寸。

**三 仿真验证结果**

**3.1 基于曼切斯特进位链的平方根选择进位加法器**

我们先仿真了1bit的基于曼切斯特进位的加法器电路，初始仿真结果如下：

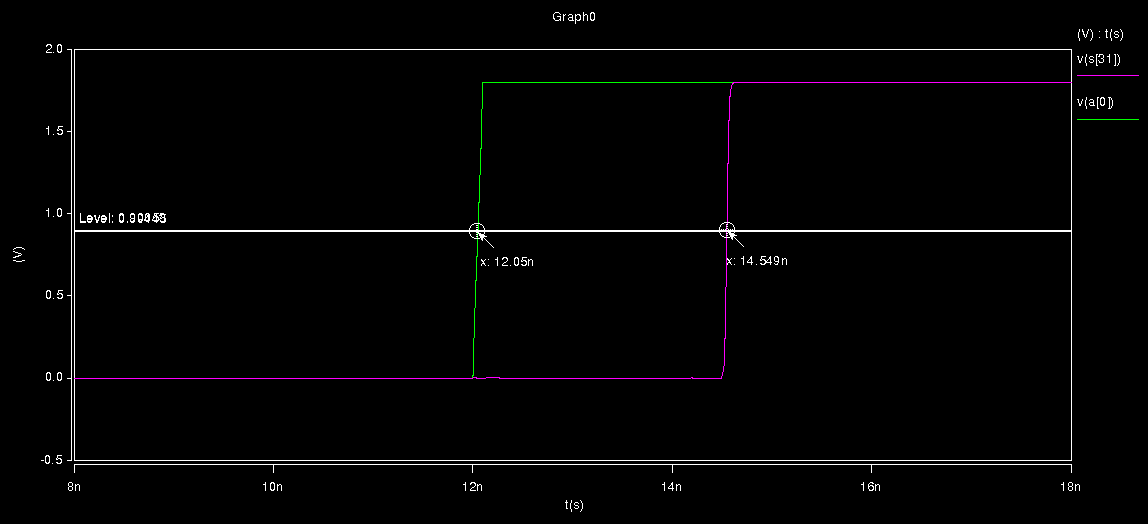


1bit传输门+曼切斯特加法器延时

得到延迟为0.243ns，接下来我们将1bit全加器组合成为32bit加法器，其中平方根选择进位结构选择2+4+5+7+8的结构。具体结构图如下：

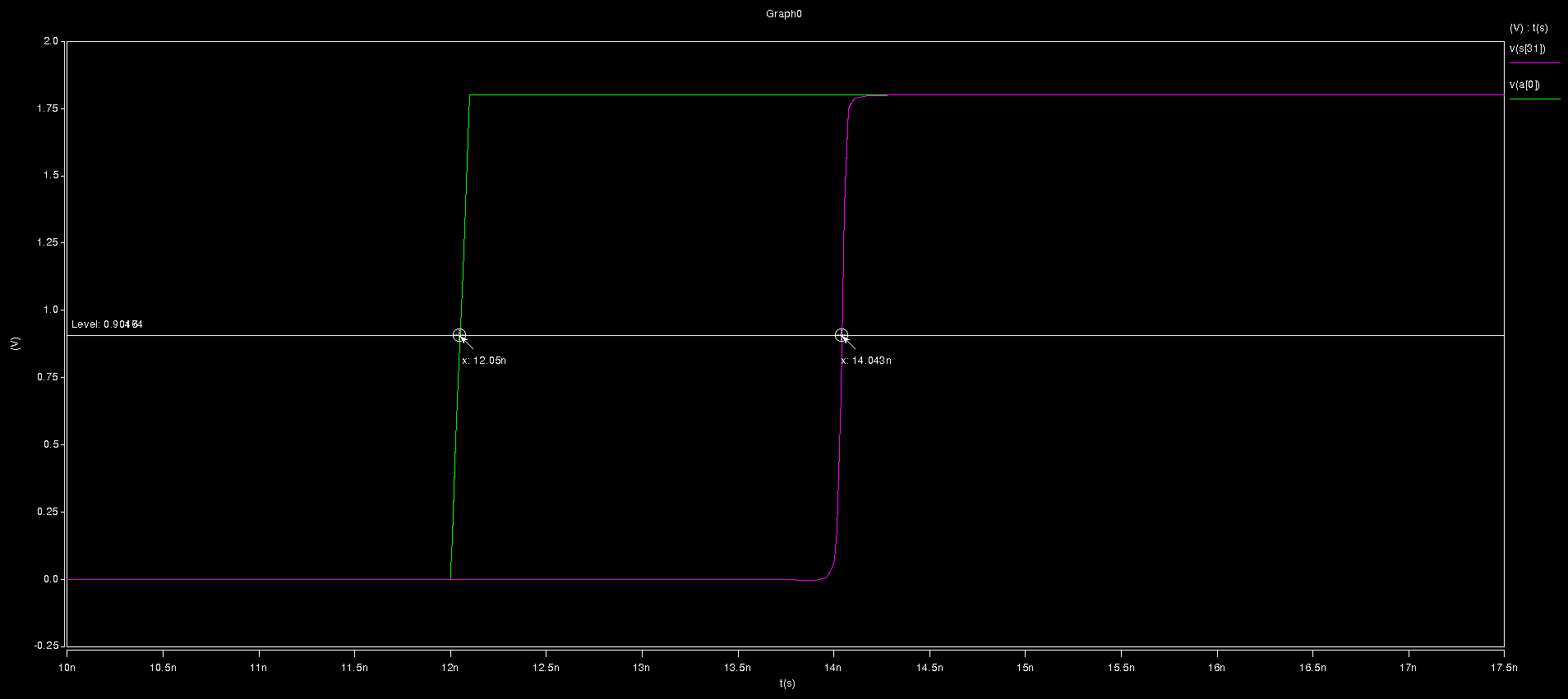


经过仿真我们得到32bit加法器总最长延时为2.499ns：



基于曼切斯特进位的32bit加法器延时

我们测试了600组随机数据，得到结果均正确，测试功耗为 这个功耗实在是太大了，无法接受的结果。我们找到原因，发现是因为我们并没有调整晶体管的尺寸大小，初始选择的尺寸都较大，导致延时较大，功耗也较大。于是我们更改了尺寸大小，并且在更改尺寸的过程中还发现最小尺寸可能会使一些模块无法正常驱动，最终，我们调试了较小的能正确工作的尺寸后，得到延时为1.993ns：

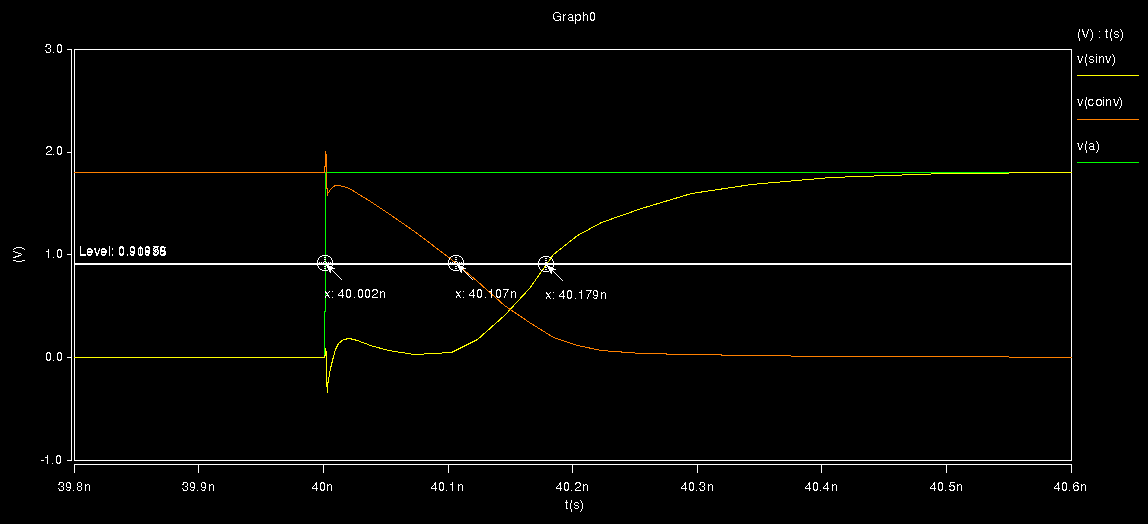


减小尺寸后的32bit加法器延时

其功耗也减小了许多，测试得到功耗为,然而功耗依然较大。我们分析原因可能是求和电路运用传输门加法器导致其延时较长，且中途需要产生许多如G，P，D等信号导致电路较为复杂，且光是进位的优化采用曼切斯特进位链反而加大了延时，可能进位直接利用传输门加法器的设计会更好，只是我们还没有尝试。

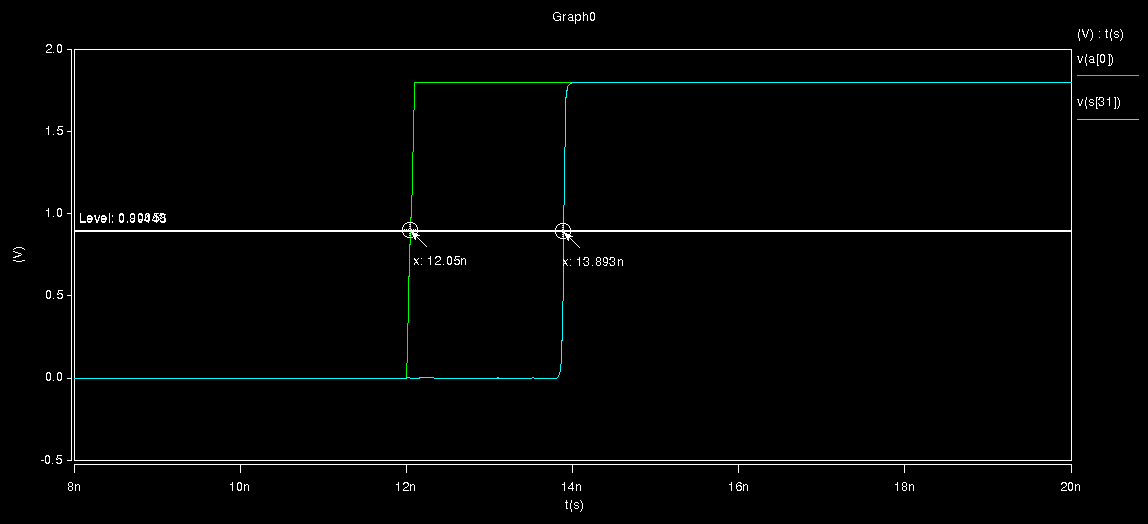
**3.2 基于镜像全加器的平方根选择进位加法器**

选择进位结构依然如前一部分图所示，我们仿真了1bit的镜像全加器，得到延时是0.177ns:



1bit镜像加法器延时

连接后仿真32bit全加器，得到最长延时为1.843ns:



32bit基于镜像全加器的加法器延时

同样仿真600组随机数，得到功耗为。发现这种电路功耗很低，延时较小。我们最终选定基于镜像全加器的平方根选择进位加法器作为我们的最终设计。

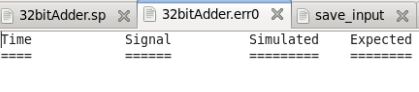
我们计算此设计的总面积如下：运用1bit镜像全加器32个，反相器34个，级间连接模块5个，二选一选择器30个，其中1bit镜像全加器面积（117\*Lmin）um，反相器（3\*Lmin）um，级间连接模块（24\*Lmin）um，二选一选择器（18\*Lmin）um，总面积为（4506\*Lmin）um（答辩ppt中有点算错了）。由Lmin=0.18um，可以计算得到总面积为811.08um。

综上，我们得到性能如下：

|  |  |
| --- | --- |
| 最长延时 | 1.843ns |
| 平均功耗 | 324.8uW |
| 面积 | 811.8um |

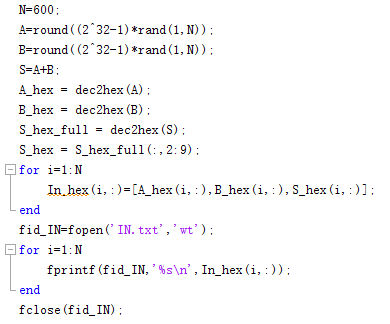
**3.3 随机数的产生与验证**

因为32位的加法较为复杂，我们不可能人工验证，于是我们采用了数字矢量文件的方式，先用matlab产生随机数和他们的和，然后hspice可以将仿真得到的结果和理论结果进行比较，最后输出一个.err0文件。如果没有错误，.err0文件的输出是这样的：

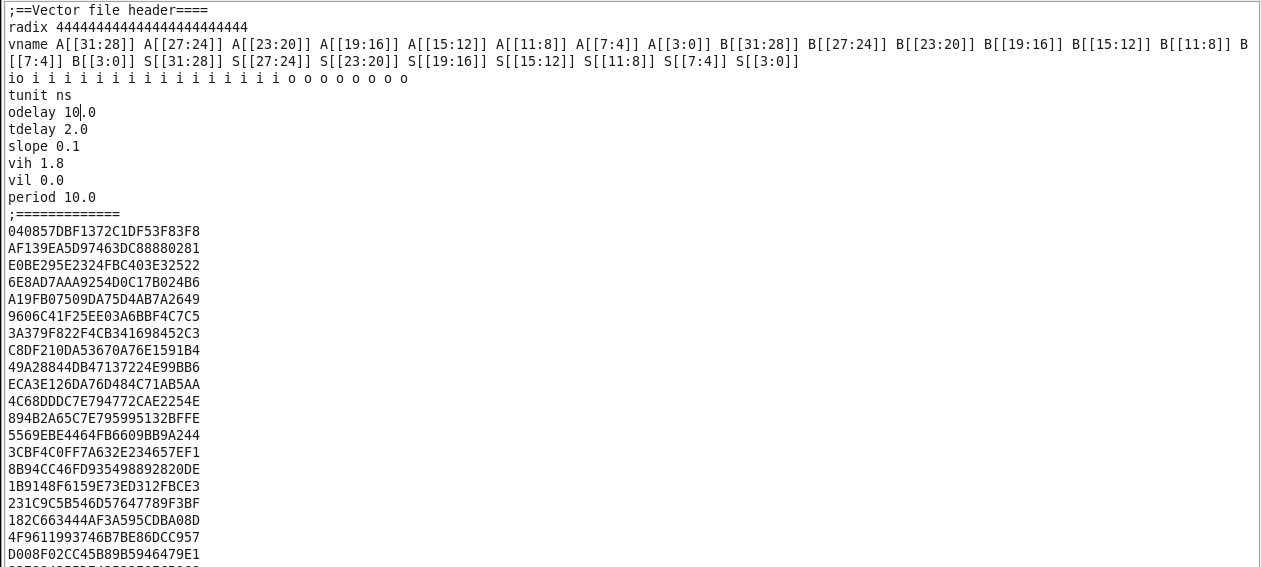


如果有错误，则会输出错误的位数，其仿真值和理论值。这样大大减小了我们验证电路正确性的时间。

matlab产生随机数及其和的代码如下：



数字矢量文件如下：



**四 设计心得**

通过这次大作业的练习让我们充分理解了加法器的运行原理，并在主动设计的过程中发现问题，并为减小延时和功耗进行电路，晶体管尺寸上的改进。这次大作业让我认识到设计一个具体的系统还是不容易的，我和我的组员经常debug到凌晨，而中途我的一场重病也严重拖延了我们小组的设计进程，以至于没有太多的时间再进行进一步的优化设计。我分析我们还可能优化的点如下：

1 对于传输门+曼切斯特加法器，我们可以全部采用传输门的形式。

2 对于镜像加法器，我们并没有像其他组一样查阅相关的文献，可能可以通过文献查阅获得更好的设计。

最后，我要特别感谢吴老师在我们做大作业的过程中对我们的细心指导和不辞辛劳的为我们组答疑解惑，在老师的帮助下我们组的设计得以优化和完善。

附：

文件夹中文件：

32bitAdder\_Mirror.sp 32bit镜像加法器

32bitAdder\_Manchester.sp 32bit曼切斯特加法器

digital\_file\_IN.vec 数字矢量输入文件