微处理器设计第一次作业

微硕21 方佳豪 2021211066

**设计目标**：32位有/无符号数加/减法器设计，要求延迟delay尽可能断，给出延迟，功耗，面积报告

**设计语言**：verilog

**工艺库**：umc18 “tt\_1v8\_25c.db”等

**设计思路：**

加法器设计的关键是解决进位关键路径的延迟，课上我们学到了一些特殊的加法器结构，来减小进位路径的延迟。对数超前进位加法器在逻辑结构上具有优越性，因此本设计主要采用该结构进行电路设计。超前进位加法器每组采用4比特运算。

1. 定义运算

进位产生：

进位传播：

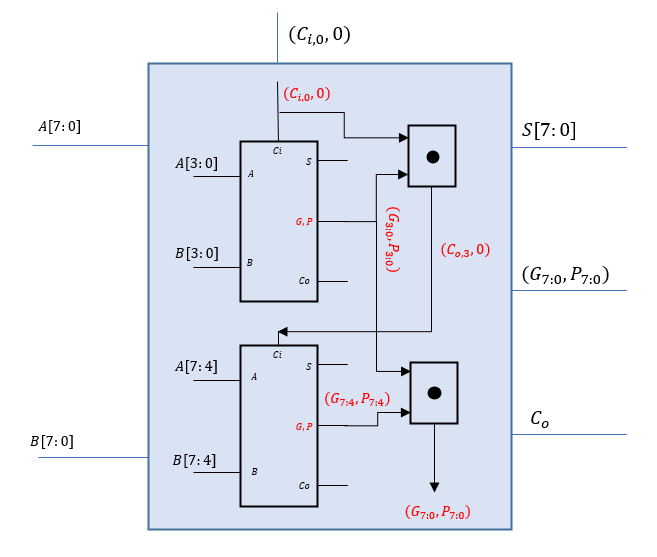
进位

和

点乘运算

进位的表示：

考虑到要实现溢出的判断，需要引出第31比特和32比特的进位信号。因而结构并非完全规整。

1. 整体结构

采用递归式设计，如右图所示结构：包含4比特一组的超前进位加法器和点乘运算模块。并且这种模块以比特拓展的方式进行递归，直到32比特。具体实现逻辑参见代码。

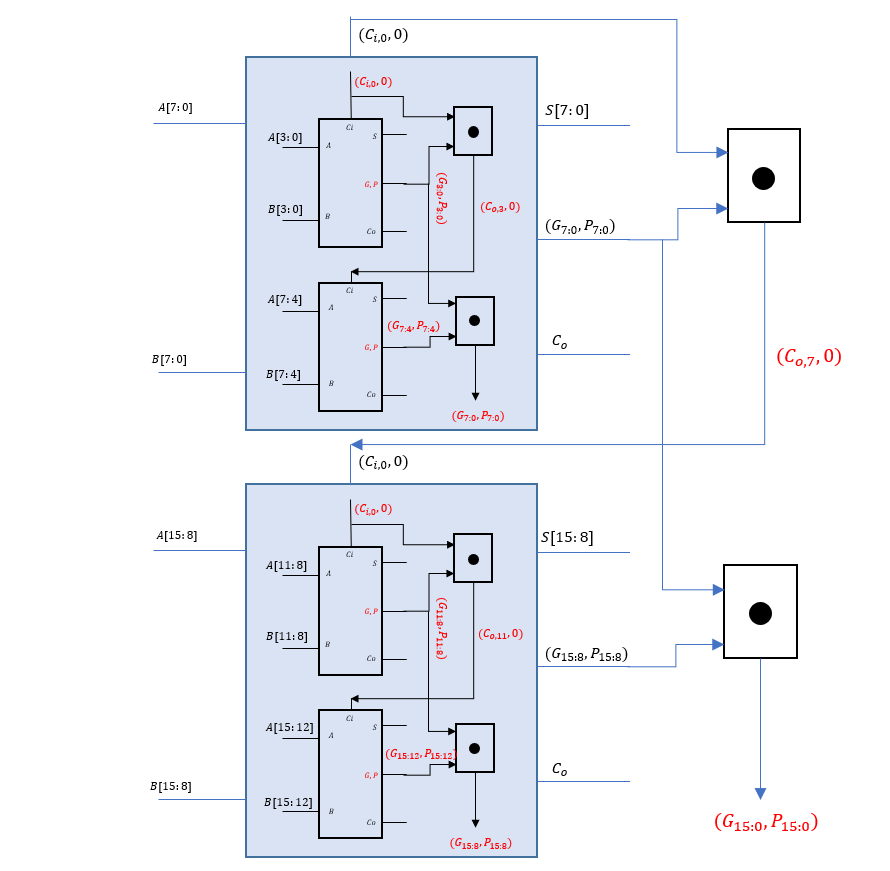
值得注意的是超前进位加法器中的代表的并非模块最高位的进位输出，而是输入。这是为做溢出判断专门引出的pin。每个加法器模块的进位只影响到S的输出，不影响G和P的生成。

整个32比特加法器共需迭代3次，存在3层点乘运算，即3层点乘即为关键路径。最后一级S也是与3层点乘运算有关。

本质4比特加法器已经迭代了2次，实际运算时间为logN量级，此处N=32。

溢出标志Overflow=isSign?(C31\_out^C31\_in):(C31\_out^isSub);

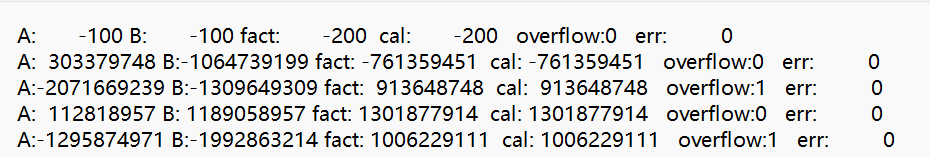
其中第31比特的进位C31\_in由上述所提到的pin直接得到，第32比特的进位输出由以下点乘运算决定：。

下图相邻两次迭代的功能拓扑图。

1. 功能验证

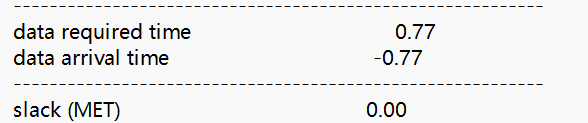
通过编写verilog的tb文件验证功能，分别对无符号运算和有符号运算进行验证：

将功能运算结果与理想结果进行对比。部分示例如下：



1. 综合结果

速度：**0.77 ns**



面积：Total cell area: **9703.108837**

功耗：

|  |  |  |  |
| --- | --- | --- | --- |
| Internal | Switching | Leakage | Total |
| 2.3204 mW | 1.2329 mW | 4.0701e+04 pW | **3.5533 mW** |

**总结：**

本次设计32位加法器，延迟为0.77ns。自己同样编写了assign c=a+b，在相同工艺库下进行综合，其延迟为0.66 ns, 表明自己的设计还是与软件自动综合生成的结果存在一定的差距。有趣的是，后者也能生成map文件。