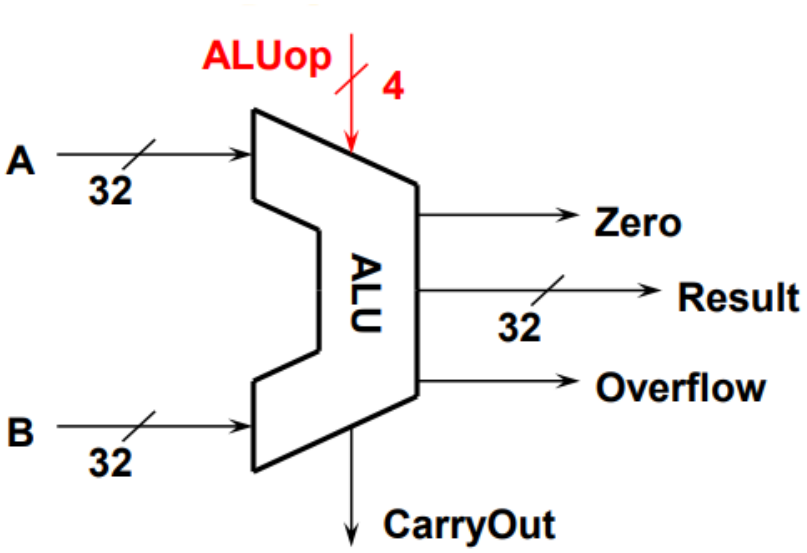
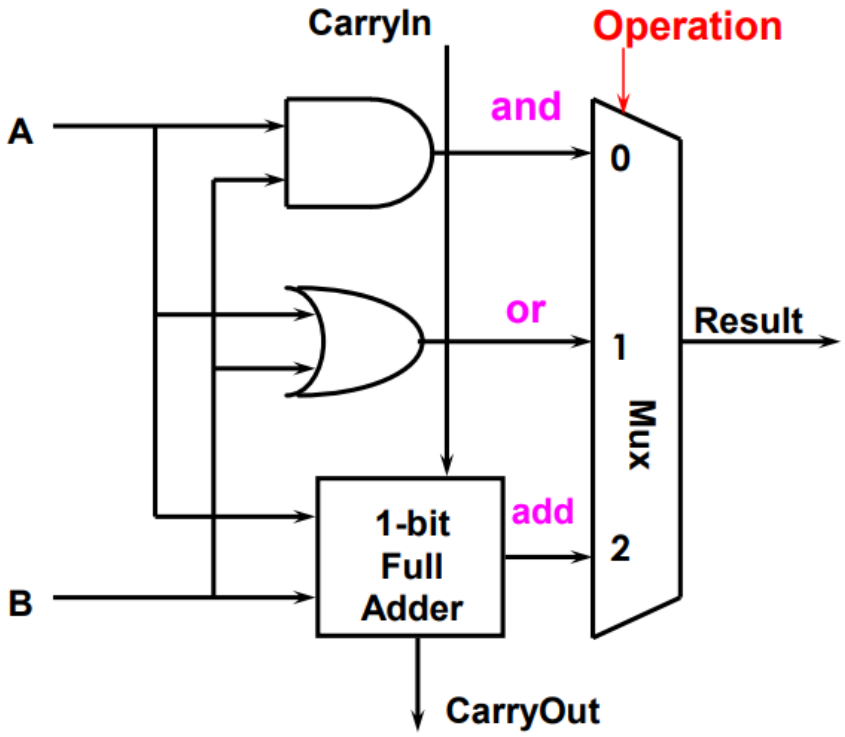
1. 算术逻辑单元(ALU)

1.1 一个32位ALU的简单模型

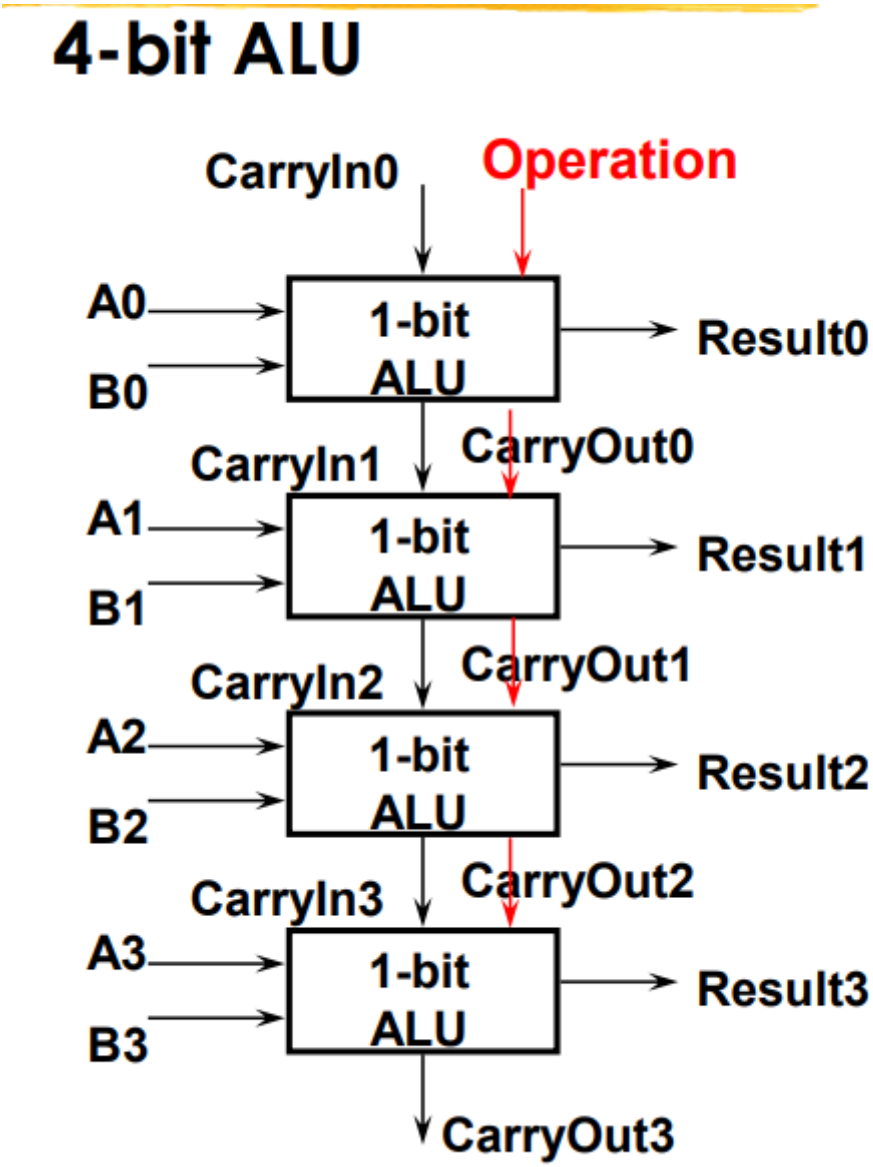


ALU Control(ALUop)	Function
0000	and
0001	or
0010	add
0110	subtract
0111	set on less than (<)
1100	nor

2．加法操作



一个1bit的ALU，通过全加器来实现加法。

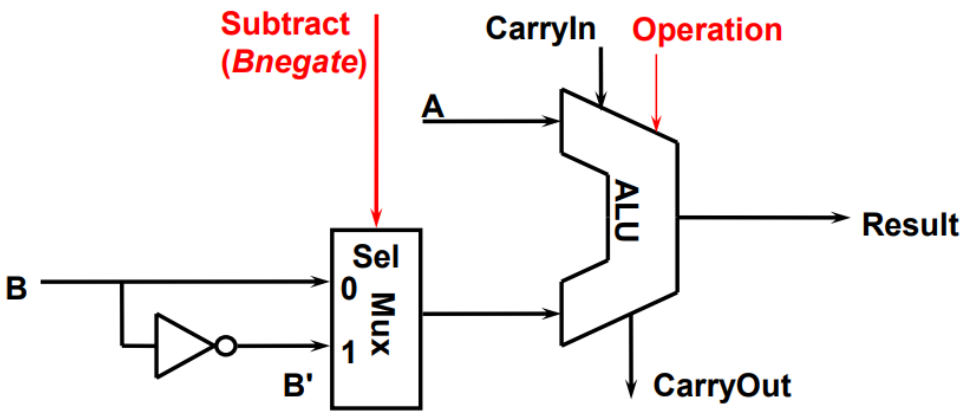


超前进位加法

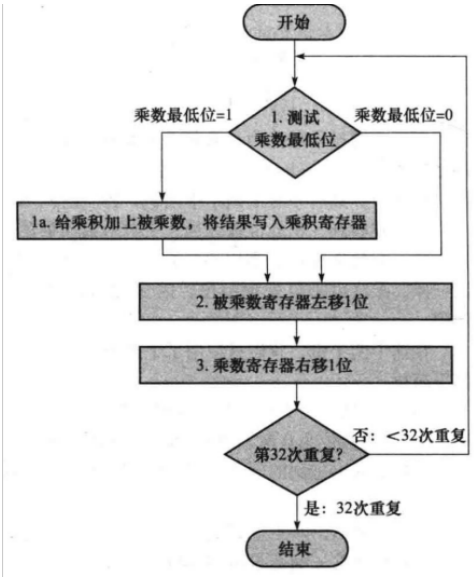
通过一定的操作，我们可以提前取得加法的进位部分，进而将串行加法器，变成块间串行，块内并行。

3．减法操作

减法的本质是加法，通过公式$-x = \overline{x} + 1$来实现。

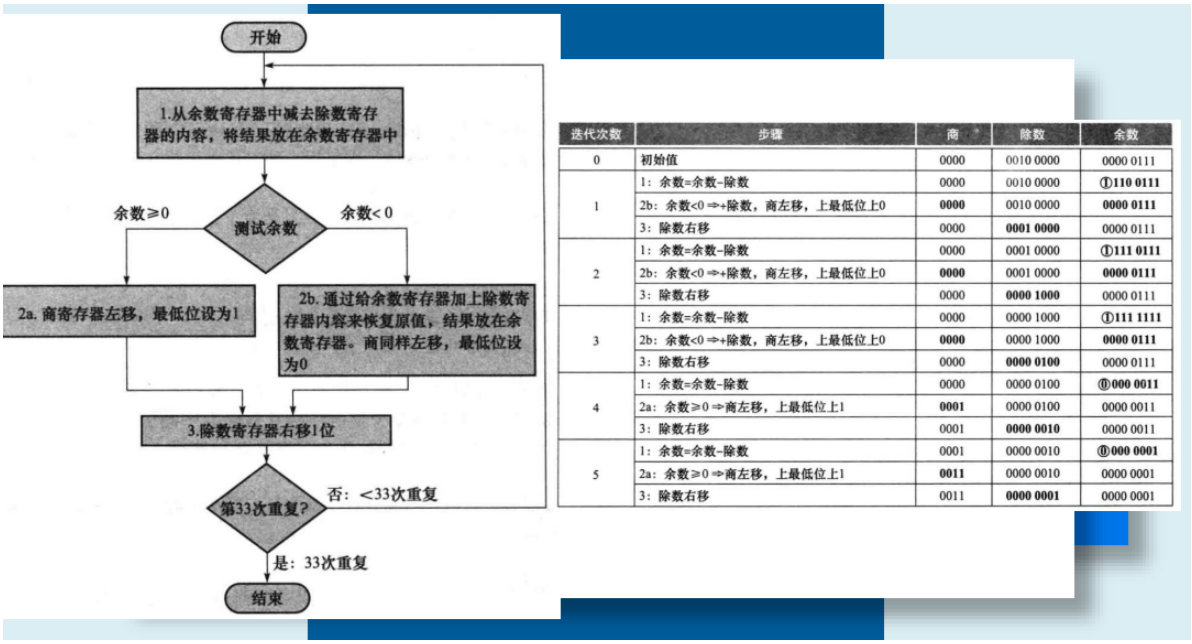


4. 乘法操作



迭代次数	步骤	乘数	被乘数	乘积
0	初始值	001①	0000 0010	0000 0000
1	1a：1⇒乘积=乘积+被乘数	0011	0000 0010	**0000 0010**
	2：左移被乘数	0011	**0000 0100**	0000 0010
	3：右移乘数	**000①**	0000 0100	0000 0010
2	1a：1⇒乘积=乘积+被乘数	0001	0000 0100	**0000 0110**
	2：左移被乘数	0001	**0000 1000**	0000 0110
	3：右移乘数	**000⓪**	0000 1000	0000 0110
3	1：0⇒无操作	0000	0000 1000	0000 0110
	2：左移被乘数	0000	**0001 0000**	0000 0110
	3：右移乘数	**000⓪**	0001 0000	0000 0110
4	1：0⇒无操作	0000	0001 0000	0000 0110
	2：左移被乘数	0000	**0010 0000**	0000 0110
	3：右移乘数	**0000**	0010 0000	0000 0110

5． 除法操作



迭代次数	步骤	商	除数	余数
0	初始值	0000	0010 0000	0000 0111
1	1：余数=余数-除数	0000	0010 0000	①**110 0111**
	2b：余数<0 ⇒+除数，商左移，上最低位上0	**0000**	0010 0000	**0000 0111**
	3：除数右移	0000	**0001 0000**	0000 0111
2	1：余数=余数-除数	0000	0001 0000	①**111 0111**
	2b：余数<0 ⇒+除数，商左移，上最低位上0	**0000**	0001 0000	**0000 0111**
	3：除数右移	0000	**0000 1000**	0000 0111
3	1：余数=余数-除数	0000	0000 1000	①**111 1111**
	2b：余数<0 ⇒+除数，商左移，上最低位上0	**0000**	0000 1000	**0000 0111**
	3：除数右移	0000	**0000 0100**	0000 0111
4	1：余数=余数-除数	0000	0000 0100	⓪**000 0011**
	2a：余数≥0 ⇒商左移，上最低位上1	**0001**	0000 0100	0000 0011
	3：除数右移	0001	**0000 0010**	0000 0011
5	1：余数=余数-除数	0001	0000 0010	⓪**000 0001**
	2a：余数≥0 ⇒商左移，上最低位上1	**0011**	0000 0010	0000 0001
	3：除数右移	0011	**0000 0001**	0000 0001

Q&A and Potpourri and Summary