1. 解：

a). 补码: 分别表示-0x70104000 和 0
b). 无符号数；分别表示 0x8FEFC000 和 0
c). 单精度浮点数；分别表示-$(1.110111111)_b \times 2^{31-127}$ 和+0.0
d). 一条 MIPS 指令：分别表示 LW R15, 0xC000(R31)以及 NOP 指令

2. 解：

a). 串行进位加法器；
每一级进位传递的延迟为 2T，因此生成 c16 需要 32T；
每一级产生结果的延迟为 3T，因此生成 s15 需要（30T+3T）=33T

b). 先行进位加法器
参照课本图 9.6 的方式搭建：组内并行、组间并行，4 位一组。
延迟共 2T（产生 p、g）+2T（产生每组 P、G）+2T（产生组间进位）+2T（产生组内进位）+ 3T（全加器逻辑）= 11T

c). 先行进位加法器快的原因是它能更快地生成第 i 位的 c 而不需要依赖于第 i-1 位的 c。

3. 解：

a). 使用多个加法器；
采用先行进位加法器两两相加，需要 2*11T=22T 延迟
b). 使用加法树及加法器。
使用加法树把四个数相加变成两个数相加，需要 2 级全加器延迟（6T），然后再使用先行进位加法器（11T）得到最后结果，因此共 6T+11T=17T 延迟。

4. 证明：

假定带符号数 x, y, x+y 都在 n 位数表示范围之内，由于求补的本质是取模运算，则它们的补码可以如下表示：$[x]_{补} = 2^{n+1}+x$, $[y]_{补} = 2^{n+1}+y$, $[x+y]_{补} = 2^{n+1}+x+y$ ，其中第 n+1 位舍弃。于是:

$[x]_{补}+[y]_{补} = 2^{n+1}+x+2^{n+1}+y = 2^{n+1} + (2^{n+1}+x+y) = 2^{n+1}+[x+y]_{补} = [x+y]_{补}$（第 n+1 位舍弃）

■

5.

```
module add16(a, b, cin, out, cout);
    input   [15:0]  a;
    input   [15:0]  b;
    input           cin;
    output  [15:0]  out;
    output          cout;
```

```
    wire [15:0] p = a|b;
    wire [15:0] g = a&b;
    wire [3:0]     P, G;
    wire [15:0]    c;

    assign c[0] = cin;

    C4 C0_3(.p(p[3:0]),.g(g[3:0]),.cin(c[0]),.P(P[0]),.G(G[0]),.cout(c[3:1]));
    C4 C4_7(.p(p[7:4]),.g(g[7:4]),.cin(c[4]),.P(P[1]),.G(G[1]),.cout(c[7:5]));
    C4 C8_11(.p(p[11:8]),.g(g[11:8]),.cin(c[8]),.P(P[2]),.G(G[2]),.cout(c[11:9]));
    C4 C12_15(.p(p[15:12]),.g(g[15:12]),.cin(c[12]),.P(P[3]),.G(G[3]),.cout(c[15:13]));
    C4 C_INTER(.p(P),.G(G),.cin(c[0]),.P(),.G(),.cout({c[12],c[8],c[4]}));

    assign cout = (a[15]&b[15]) | (a[15]&c[15]) | (b[15]&c[15]);
    assign out = (~a&~b&c)|(~a&b&~c)|(a&~b&~c)|(a&b&c);
endmodule
```

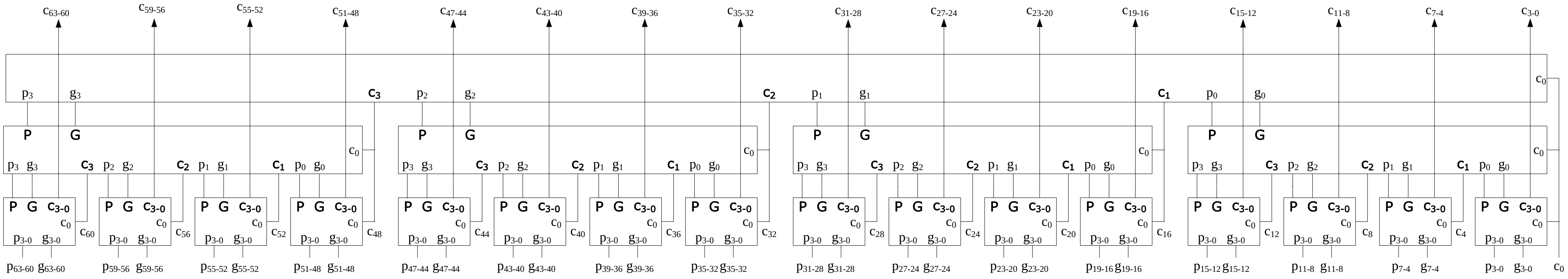
```
module C4(p,g,cin,P,G,cout)
    input    [3:0]    p, g;
    input             cin;
    output            P,G;
    output   [2:0]    cout;

    assign P=&p;
    assign G=g[3]|(p[3]&g[2])|(p[3]&p[2]&g[1])|(p[3]&p[2]&p[1]&g[0]);

    assign cout[0]=g[0]|(p[0]&cin);
    assign cout[1]=g[1]|(p[1]&g[0])|(p[1]&p[0]&cin);
    assign cout[2]=g[2]|(p[2]&g[1])|(p[2]&p[1]&g[0])|(p[2]&p[1]&p[0]&cin);
endmodule
```

6.

参数	格式			
	单精度	扩展单精度	双精度	扩展双精度
尾数位宽 P	23	>=23	52	>=52
指数最大值 Emax	127	>=127	1023	>=1023
指数最小值 Emin	-126	1-Emax	-1022	1-Emax
指数偏移量 Bias	127	Emax	1023	Emax
指数位数	8	>=8	11	>=11

浮点格式宽度	32	>32	64	>64

7.


c63-60 c59-56 c55-52 c51-48 c47-44 c43-40 c39-36 c35-32 c31-28 c27-24 c23-20 c19-16 c15-12 c11-8 c7-4 c3-0
p3 g3 c3 p2 g2 c2 p1 g1 c1 p0 g0 c0
P G
p3 g3 c3 p2 g2 c2 p1 g1 c1 p0 g0 c0
P G c3-0 c0 p3-0 g3-0
c60 c56 c52 c48 c44 c40 c36 c32 c28 c24 c20 c16 c12 c8 c4
p63-60 g63-60 p59-56 g59-56 p55-52 g55-52 p51-48 g51-48 p47-44 g47-44 p43-40 g43-40 p39-36 g39-36 p35-32 g35-32 p31-28 g31-28 p27-24 g27-24 p23-20 g23-20 p19-16 g19-16 p15-12 g15-12 p11-8 g11-8 p7-4 g7-4 p3-0 g3-0 c0

正确性证明略。

8. 答：

16 个数相加的华莱士树，按照课本上的画法，共有 14 个进位输入；最后的全加器有一个 cin；最后加法器中，C 和 S 是错位相加的，因此 C 的低位还有一个空位。共计 16 个位置。