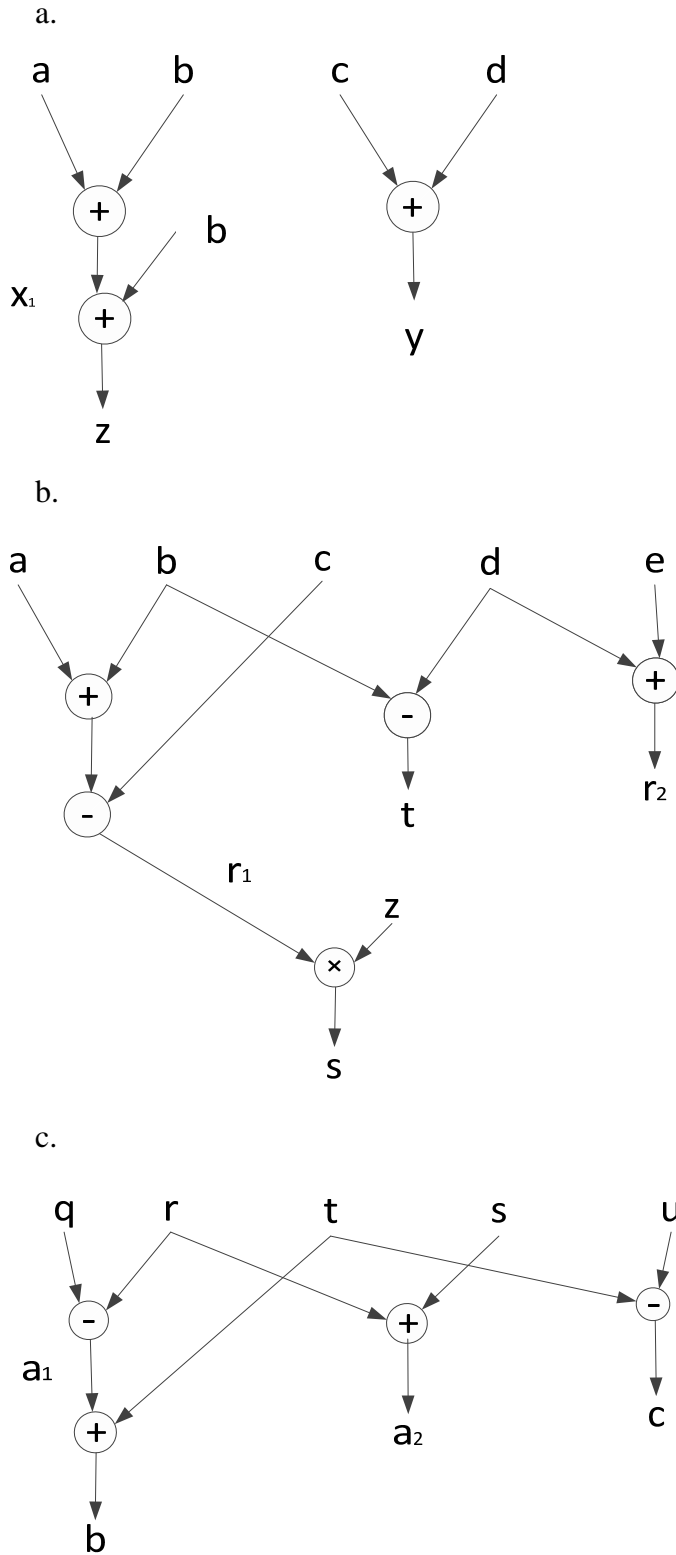


# 第五章

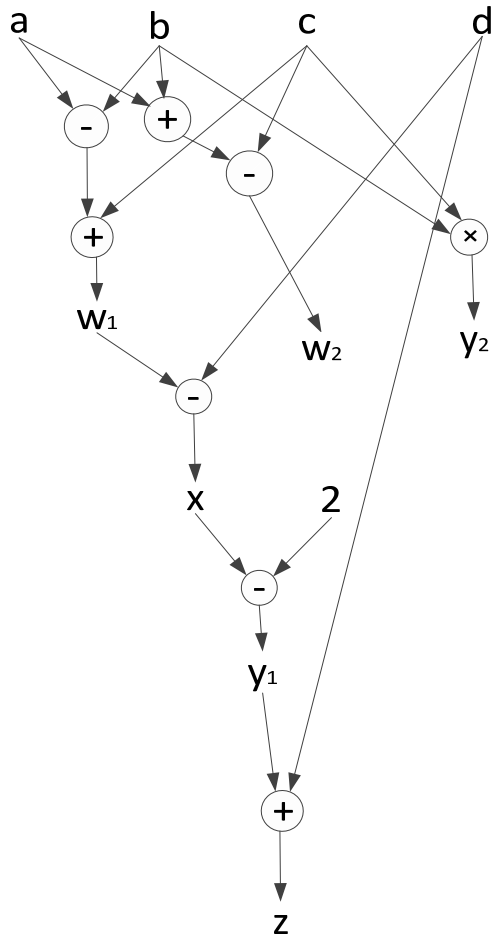
Q5-2

参考编程示例 5-3

Q5-4

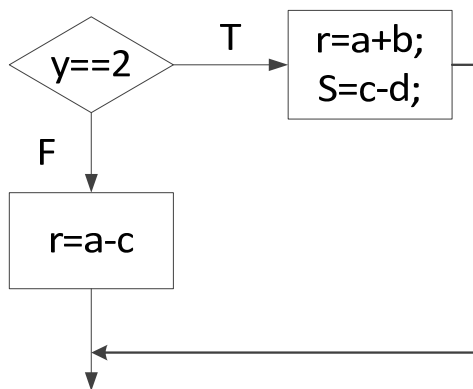


d.

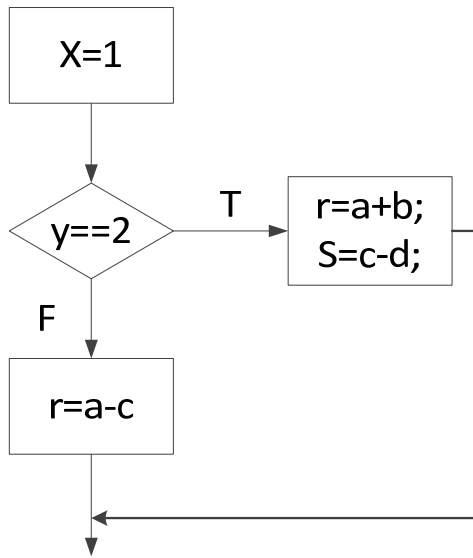


Q5-5

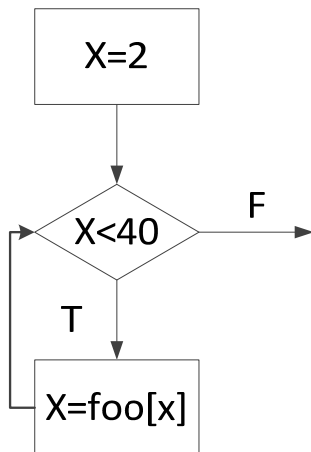
a.



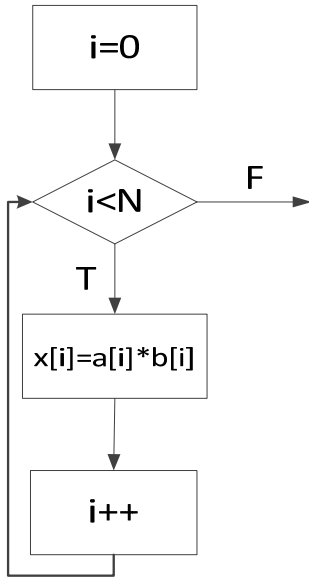
b.



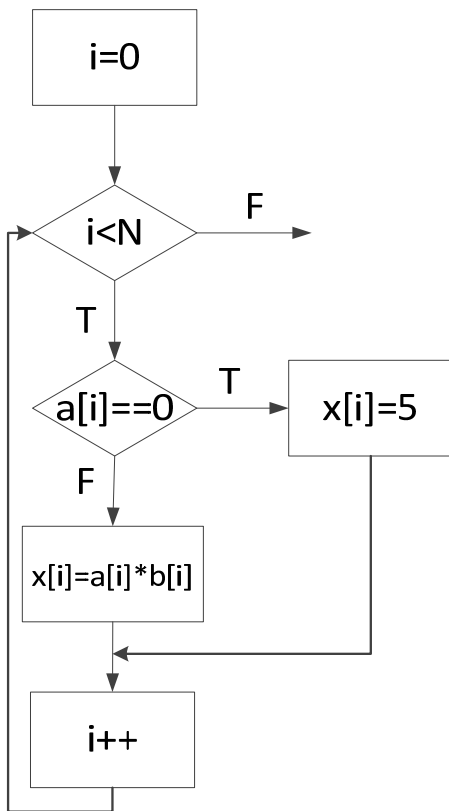
c.



d.



e.



Q5-6

a.符号表:

P1	200
P2	228

b.符号表:

p1	100
p2	108
p3	116

c.符号表

S1	200
S2	208

Q5-7

答: 可以, 外部引用是入口点的子集。

Q5-8

a 可重入, b, c 不可重入 (请查看百度百科关于可重入函数的完整定义和解释)

Q5-9

使用和改变全局变量数组, 不是可重入函数

Q5-10,11 不作要求

Q5-12

a.两次

```
for (i=0; i<15; i++)  
{  
    x[i*2] = a[i*2]*c[i*2];  
    x[i*2+1] = a[i*2+1]*c[i*2+1];  
}
```

b.四次

```
for(i=0;i<8;i++)  
{  
    x[i*4]=a[i*4]*c[i*4];  
    x[i*4+1]=a[i*4+1]*c[i*4+1];  
    x[i*4+2]=a[i*4+2]*c[i*4+2];  
    x[i*4+3]=a[i*4+3]*c[i*4+3];  
}
```

```
}
```

Q5-13

a、for (i=0; i<N; i++)

```
{
```

```
    z[i] = a[i] + b[i];
```

```
    w[i] = a[i] - b[i];
```

```
}
```

b、c 不要求

Q5-14 不能使用“代码移出”，可以使用“归纳变量消除”

```
zbinduct = 0;
```

```
ainduct = 0;
```

```
for (i=0; i<N; i++) {
```

```
    for (j=0; j<M; j++) {
```

```
        *(zptr + zbinduct) = *(aptr + ainduct) × *(bptr + zbinduct);
```

```
        zbinduct++;
```

```
    }
```

```
    ainduct++;
```

```
}
```

Q5-15

- 最少需四个寄存器。
- 最少需四个寄存器。
- 最少需要四个寄存器。
- 最少需要六个寄存器。

Q5-16

- 将第二句和第三句对换，最少需要的寄存器就变为三个。
- 已为最少，不需改动。
- 已为最少，不需改动。
- 已为最少，不需改动。

Q5-17

**题目补充：cache 为直接映射方式**

提示：该高速缓存共有 256 行，每个高速缓存行长为 4 个字，数组每个元素大小为 1 个字。

- a.  $a[0][0]$ 与  $x[0][0]$ 的低 12 位地址相同;
- b.  $a[0][0]$ 与  $x[0][0]$ 的低 12 位地址相比小 1 个字;
- c.  $a[0][0]$ 与  $x[0][0]$ 的低 12 位地址相差大于等于 4 个字;