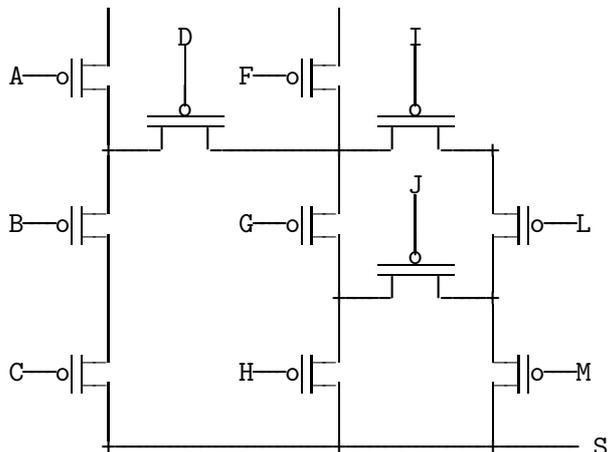


Que valent CF , OF , ZF et SF après les opérations $4+5$, $15-2$, $5+9$, $2-7$, $3-8$, $11-12$, $9+12$, $1+5$, $0-11$, $15-12$, $2-0$, $8-2$, $5+14$ et $12+14$ sur des nombres codés sur 4 bits.

Complétez le bas du schéma avec autant de transistors.



```
f: sub r3,r2,r5
   loadimm16 r3,1
   xor r4,r4,r4
l1: sub r0,r3,r0
   jc l2
   load r1,r6
   sub r6,r2,r7
   mul r6,r6,r6
   add r6,r4,r6
   sub r7,r5,r7
   movccbe r6,r4 // if( <= ) r4=r6 non signé
   add r1,r3,r1
   jmp l1
l2: mov r4,r0
   ret
```

Donnez un équivalent simple en C de la fonction f.

Que vaut x dans `int t[]={3,6,1,5,3,4,2}`, `x=f(7,t,3,5)`;

Donnez une formule donnant la valeur de $f(n,t,min,max)$. A défaut dites en une phrase ce que calcule f.

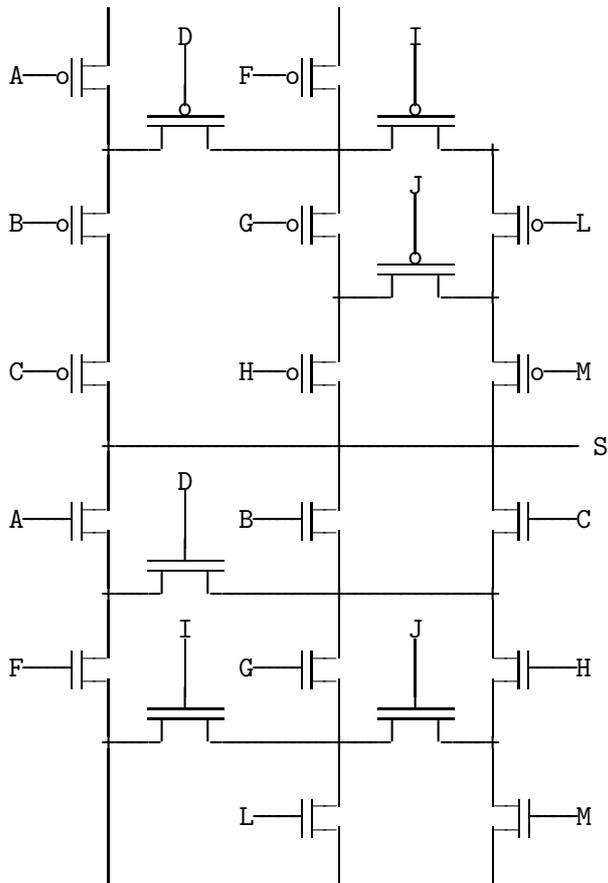
Même question si on remplace `movccbe` par `movccb`.

Redonnez la valeur de x et la formule de $f(n,t,min,max)$ si on supprime la ligne `mul r6,r6,r6`.

Ecrivez en C puis en assembleur la fonction `int g(int n, int *t, int min, int max)`; qui rend la somme des cubes des nombres qui parmi les n premiers éléments du tableau t ne sont pas compris dans l'intervalle $[min,max]$.

Corrigé

non signé	signé	CF	OF	ZF	SF
4+ 5= 9	4+ 5=-7	0	1	0	1
15- 2=13	-1- 2=-3	0	0	0	1
5+ 9=14	5+ -7=-2	0	0	0	1
2- 7=11	2- 7=-5	1	0	0	1
3- 8=11	3- -8=-5	1	1	0	1
11- 12=15	-5- -4=-1	1	0	0	1
9+ 12= 5	-7+ -4= 5	1	1	0	0
1+ 5= 6	1+ 5= 6	0	0	0	0
0- 11= 5	0- -5= 5	1	0	0	0
15- 12= 3	-1- -4= 3	0	0	0	0
2- 0= 2	2- 0= 2	0	0	0	0
8- 2= 6	-8- 2= 6	0	1	0	0
5+ 14= 3	5+ -2= 3	1	0	0	0
12+ 14=10	-4+ -2=-6	1	0	0	1



```

f:  sub  r3,r2,r5 //          r0   r1   r2           r3   r4   r5           r6   r7
    loadimm16 r3,1//int f(int n,int*t,int min,int max)//1 s   d=       x   e=
    xor   r4,r4,r4 //{ int s=0;           //          max-min *t   *t-min
11: sub  r0,r3,r0
    jc   l2          // while(n--)          int f(int n, int*t,
    load r1,r6      // { int x=*t,          unsigned min, int max)
    sub  r6,r2,r7 //          e=x-min;      { while(n--)
    mul  r6,r6,r6 //          x*=x;          { int x=*t++;
    add  r6,r4,r6 //          x+=s;          if(x-min<=max-min) s+=x*x;
    sub  r7,r5,r7 //          }
    movccbe r6,r4 //          if(e+0u<=d+0u) s=x; return s;
    add  r1,r3,r1 //          t++;          }
    jmp  l1          // }
12: mov  r4,r0      // return s;
    ret                //}

```

$$x = 3^2 + 5^2 + 3^2 + 4^2 = 59$$

$$f(n, t, min, max) = \sum_{\substack{0 \leq i < n \\ min \leq t[i] \leq max}} t[i]^2$$

C'est la somme des carrés des nombres contenus dans le tableau t de dimension n qui sont compris entre min et max .

$$x = 3^2 + 3^2 + 4^2 = 34$$

$$f(n, t, min, max) = \sum_{\substack{0 \leq i < n \\ min \leq t[i] < max}} t[i]^2$$

$$x = 3 + 5 + 3 + 4 = 15$$

$$f(n, t, min, max) = \sum_{\substack{0 \leq i < n \\ min \leq t[i] \leq max}} t[i]$$

```

int g(int n, int*t, unsigned min, int max)
{ while(n--)
  { int x=*t++;
    if(x-min>max-min) s+=x*x*x;
  }
return s;
}

```

Dans la fonction f donnée dans l'énoncé on remplace `movccbe` par `movcca` et la ligne

```

mul  r6,r6,r6
par
mul  r6,r6,r8
mul  r6,r8,r6

```

Barème

1) 7pt=14x0.5pt

Chaque opération: 0 ou 0.5pt.

2) 4pt

-0.5pt pour toute erreur : Il manque 1 ou plusieurs !. Chaque lettre (commande) manquante ou en trop ou mal placée.

3) 6pt

f en C 1.5 pt-0.5pt par erreur comme argument manquant, test de boucle faux, incrément oublié

59 0.75pt

formule 0.75pt

34 0.75pt

formule 0.75pt

15 0.75pt

formule 0.75pt

4) 3pt

g en C 1 pt

movccb 1 pt

2 mul pour x^3 1 pt

-0.5pt pour chaque erreur.