ARQUITECTURA SISTEMAS DISTRIBUIDOS. 3º GTI
REPASO PROBLEMAS
SOLUCIONES PROBLEMAS 1, 2, 3, 4, 5 Y 6

1) Desarrollo de la solución para el ejemplo de Pentium 4 a 1.7 GHz:

$$S_{P4\ respecto\ de\ P3} = \frac{T_{P3}}{T_{P4}} = \frac{\cancel{N}\times CPI_{P3}\times\tau_{P3}}{\cancel{N}\times CPI_{P4}\times\tau_{P4}} = \frac{CPI_{P3}}{CPI_{P4}}\times\frac{\tau_{P3}}{\tau_{P4}} = \frac{CPI_{P3}}{CPI_{P4}}\times\frac{f_{P4}}{f_{P3}} = 1.26 \xrightarrow{de\ donde} \frac{CPI_{P3}}{CPI_{P4}} = \frac{1.26}{\frac{f_{P4}}{f_{P3}}} = \frac{1.26}{\frac{1.7}{1}} = \frac{1.26}{1.7}, pero\ nos\ piden\ \frac{CPI_{P4}}{CPI_{P3}} = \frac{1.7}{1.26} = 1.349$$

El resto de casos es similar.

2) Suponemos r1 = &x[0]. rfin = &y[0]. Todos los vectores están contiguos en memoria, uno detrás de otro. Primero el vector x, luego el y, luego el z. Cada elemento es un float (4 bytes), por lo que cada vector ocupa 100*4 bytes en memoria.

```
lf fy,400(r1)
lf fz,800(r1)
divf fx,fy,fz
sf 0(r1),fx
addi r1,r1,4
slt r3,r1,rfin
bnez r3,loop
```

Mirar cronogramas en hoja Excel para el resto de apartados.

3) Suponemos r1 = &a[2]. rfin = &b[0]. Todos los vectores están contiguos en memoria, uno detrás de otro. Primero el vector a, luego el vector b. Cada elemento es un float (4 bytes) por lo que cada vector ocupa 1024*4 bytes en memoria. f1 contiene el valor de a[i], f2 el de a[i-2] y f3 el de b[i-1]

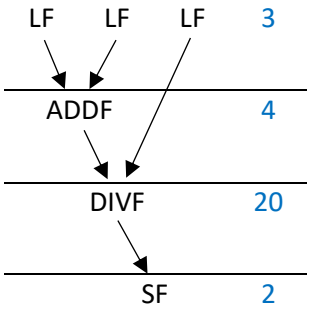
```
lf f1,0(r1)
lf f2,-8(r1)
lf f3,4092(r1)
addf f1,f1,f2
divf f1,f1,f3
sf 0(r1),f1
addi r1,r1,4
slt r3,r1,rfin
bnez r3,loop
```

Reordenando para que el mayor número posible de instrucciones se ejecute antes del divf…

```
lf f1,0(r1)
lf f2,-8(r1)
lf f3,4092(r1)
addf f1,f1,f2
addi r1,r1,4
slt r3,r1,rfin
divf f1,f1,f3
sf -4(r1),f1
bnez r3,loop
```

4) Usaremos la primera versión del bucle, aunque para este estudio da igual en realidad cuál se use.

Para una única iteración tenemos:



Total: **29**

Para muchas iteraciones, desenrollamos el bucle obteniendo esto. Marcamos las dependencias reales entre iteraciones.

```
a[2] = (a[2] + a[0]) / b[1];
a[3] = (a[3] + a[1]) / b[2];
a[4] = (a[4] + a[2]) / b[3];
a[5] = (a[5] + a[3]) / b[4];
a[6] = (a[6] + a[4]) / b[5];
a[7] = (a[7] + a[5]) / b[6];
a[8] = (a[8] + a[6]) / b[7];
a[9] = (a[9] + a[7]) / b[8];
...
```

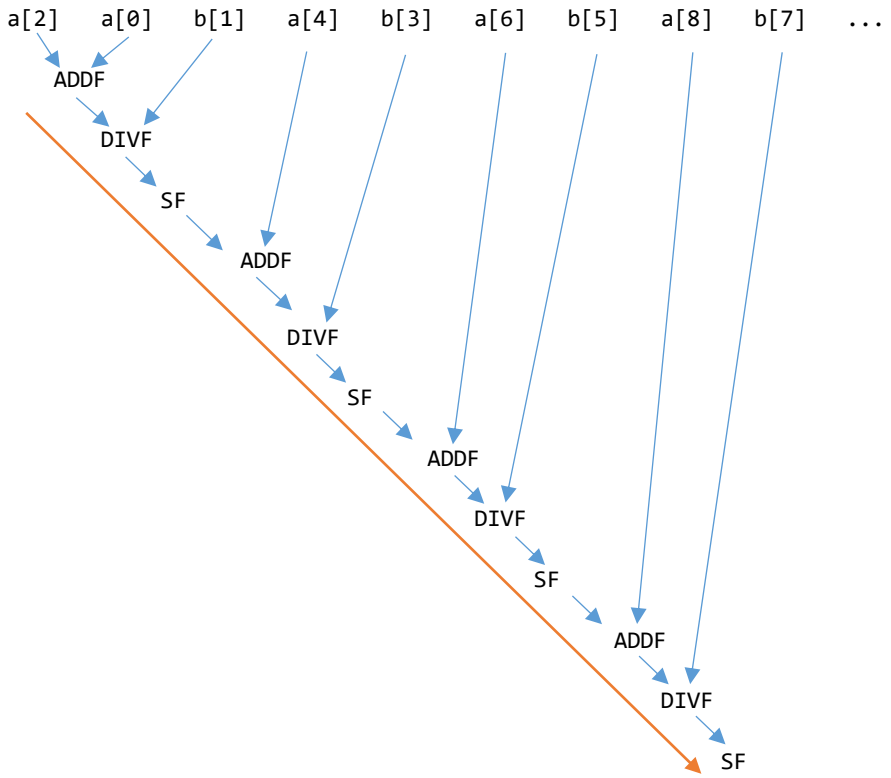
Podemos observar que las dependencias reales afectan a una de cada dos iteraciones. Las iteraciones pares tienen dependencias entre sí, al igual que las impares entre sí, PERO las iteraciones pares no tienen dependencia con las impares, lo cual nos permite reescribir este desenrollado de esta otra forma:

```
a[2] = (a[2] + a[0]) / b[1];
a[4] = (a[4] + a[2]) / b[3];
a[6] = (a[6] + a[4]) / b[5];
a[8] = (a[8] + a[6]) / b[7];
...
```

```
a[3] = (a[3] + a[1]) / b[2];
a[5] = (a[5] + a[3]) / b[4];
a[7] = (a[7] + a[5]) / b[6];
a[9] = (a[9] + a[7]) / b[8];
...
```

Cada iteración de cada uno de estos dos sub-bucles sigue durando 29 ciclos, pero ahora una iteración impar puede ejecutarse en paralelo con una par, con lo que la duración percibida de cada iteración es la mitad: $\frac{29}{2}$

El camino crítico, en cada sub-bucle sería (por ejemplo, para el de la izquierda):



5) Suponemos r1 = &y[0]. rfin = &x[0]. Todos los vectores están contiguos en memoria, uno detrás de otro. Primero el vector y, luego el vector x. Cada elemento es un float (4 bytes) por lo que cada vector ocupa 1024*3*4 bytes en memoria.

```
lf fy,0(r1)
lf fx,12288(r1)
mulf fx,fa,fx
addf fy,fx,fy
sf 0(r1),fy
addi r1,r1,4
slt r3,r1,rfin
bnez r3,loop
```

SAXPY original: N_{orig} = 1024*3*8 instrucciones = 24576 instrucciones

De esas 8 instrucciones, 5 son útiles y 3 de overhead del bucle.

SAXPY desenrollado 3 veces: N_{desen} = 1024*(5 instrucciones útiles * 3 iteraciones + 3 de overhead) = 18432 instrucciones

$$S = \frac{t_{saxpy_{orig}}}{t_{saxpy_{desen}}} = \frac{N_{orig} \times CPI_{orig} \times \tau}{N_{desen} \times CPI_{desen} \times \tau} = \frac{24576}{18432} \times \frac{1+0{,}15+\frac{1}{8}\times 1{,}5}{1+0{,}15+\frac{1}{18}\times 1{,}5}$$

6) Ancho de banda, calculado para 1 iteración del bucle original:

$$AB_{1\_iteracion} = \frac{Bytes\ accedidos}{duracion\ iteración} = \frac{\sum(numero\ de\ accesos \times bytes\ en\ cada\ acceso)}{N_{iteracion\_orig} \times CPI_{orig} \times \tau} = \frac{3\ accesos \times 4\ bytes/acceso}{8\ instrucciones \times \left(1+0{,}15+\frac{1}{8}\times 1{,}5\right) \times \frac{1}{2\times 10^9}}$$

Ancho de banda, calculado para 1 iteración del bucle desenrollado:

$$AB_{1\_iteracion\_desenrollada} = \frac{9\ accesos \times 4\ bytes/acceso}{18\ instrucciones \times \left(1+0{,}15+\frac{1}{18}\times 1{,}5\right) \times \frac{1}{2\times 10^9}}$$