

UNIVERSIDAD CARLOS III DE MADRID

# Tema 11. Predicción de Saltos

---

Departamento de Ingeniería de Sistemas y Automática

RAÚL PÉRULA MARTÍNEZ  
LUIS ENRIQUE MORENO LORENTE  
ALBERTO BRUNETE GONZALEZ  
CESAR AUGUSTO ARISMENDI GUTIERREZ  
DOMINGO MIGUEL GUINEA GARCIA ALEGRE  
JOSÉ CARLOS CASTILLO MONTOYA



Universidad  
Carlos III de Madrid



Esta obra se publica bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartidaIgual 3.0 España.



## Ejercicio 1

Suponiendo que el siguiente código en C:

```
for (i=0; i<100; i++) {  
    if(B[i] <= 0)  
        A[i] = B[i] + C[i];  
}
```

Se tiene en un cierto lenguaje ensamblador:

```
      L0:  ld    r4, #100  
          ld    r1, DIR_A  
          ld    r2, DIR_B  
          ld    r3, DIR_C  
1  L1:  ld    f2, 0(r2)  
2      blt   f2, 0, L2  
3      ld    f4, 0(r3)  
4      addd  f6, f4, f2  
5      store 0(r1), f6  
6  L4:  add   r1, r1, #4  
7      add   r2, r2, #4  
8      add   r3, r3, #4  
9      sub   r4, r4, #-4  
10     blt   r4, 0, L1
```

Donde se lista en primer lugar el operando de destino y a continuación los operandos fuente, y en el caso de las instrucciones de salto se listan primero los operandos a ser comparados y después la dirección de salto.

Se pide:

1. Dado el esquema de la arquitectura del procesador DLX, indicar en qué etapa o etapas del cauce y qué conexiones y elementos hardware serían necesarias para que dicha arquitectura pueda realizar predicción dinámica de saltos (razonarlo).
2. Obtener el cronograma de ejecución para el ejemplo en ensamblador suponiendo que la arquitectura del procesador es la DLX, para el caso que puede realizar predicción dinámica de saltos y para el caso de que no se use predicción dinámica de saltos. (Asumimos que  $0(r2) = 1$ )
3. Muestre el cronograma para el caso anterior, suponiendo que la máquina tiene predicción dinámica de saltos pero que dicha predicción falla. Razone las acciones que se producen en la arquitectura.



## Ejercicio 2

Suponiendo que el siguiente código en C:

```
for (i=0; i<100; i++) {  
    A[i] = B[i] + C[i];  
}
```

Se tiene en un cierto lenguaje ensamblador:

```
      L0:  ld    r4, #100  
          ld    r1, DIR_A  
          ld    r2, DIR_B  
          ld    r3, DIR_C  
1     L1:  ld    f2, 0(r2)  
2          ld    f4, 0(r3)  
3          addd f6, f4, f2  
4          store 0(r1), f6  
5          add  r1, r1, #4  
6          add  r2, r2, #4  
7          add  r3, r3, #4  
8          sub  r4, r4, #-4  
9          blt  r4, 0, L1
```

Donde se lista en primer lugar el operando de destino y a continuación los operandos fuente, y en el caso de las instrucciones de salto se listan primero los operandos a ser comparados y después la dirección de salto.

Se pide:

1. Obtener el cronograma de ejecución para dos iteraciones sucesivas pero suponiendo que la máquina DLX dispone de una caché para predicción dinámica de saltos (BTAC).
2. Muestre como sería el cronograma para el apartado anterior en el caso de que la máquina DLX utilizase planificación dinámica basada en el algoritmo de Tomasulo. La arquitectura es la siguiente:
  - 1 UF add 2 ciclos,
  - 1 UF mul 2 ciclos,
  - 1 UF div 2 ciclos no segmentados.
  - 2 estaciones de reserva por unidad funcional.