



IIC2343 - Arquitectura de Computadores (II/2025)

Ayudantía 15

Ayudantes: Daniela Ríos (danielaarp@uc.cl), Alberto Maturana (alberto.maturana@uc.cl), Tomás López Massaro (tomas.lopezm20@uc.cl)

Pregunta 1: Preguntas Conceptuales

- (a) Describa cómo se puede modificar el computador básico con *pipeline* para que las predicciones de salto erróneas pierdan dos ciclos en vez de tres.
- (b) Considere el siguiente código escrito para el assembly del computador básico con *pipeline*:

```
ADD A,B
MOV (var0), A
MOV (var1), B
```

1. Indique con un diagrama los *hazards* que se generan al ejecutar el código anterior.
 2. Indique que sucede al intercambiar el orden de la segunda y tercera línea del código. ¿Cambia el resultado final? En caso de existir *hazards*, indíquelos con un diagrama.
 3. Para los dos casos anteriores, en caso de existir *hazards*, indique cómo los detectan las *forwarding units* correspondientes, especificando las señales de control participantes.
- (c) Estime, en términos de N , el número de ciclos que toma la ejecución de un programa de N instrucciones en el computador básico con *pipeline* si en este:
- El 10 % de las instrucciones son de salto y, en el 25 % de estas, el salto se realiza.
 - El 30 % de las instrucciones son de escritura de memoria y, el 40 % de estas, escribe datos obtenidos en la instrucción anterior o previa a la anterior.
 - El 20 % de las instrucciones son de lectura de memoria y, en el 70 % de estas, se usa el dato leído en la siguiente instrucción.

Pregunta 2: Paralelismo a nivel de instrucción (ILP)

Suponga que se modifica el computador básico con *pipeline* para reincorporar las instrucciones de manejo de *stack*: PUSH A/B y POP A/B. Indique, con ejemplos, los tipos de *hazard* que podrían surgir de las instrucciones añadidas en casos distintos, explicando en cada ejemplo la etapa

en la que surge la dependencia. Se considerarán “casos distintos” si se detectan con combinaciones de señales distintas.

Pregunta 3: Paralelismo a nivel de instrucción (ILP) [P3-I3-2024-1]

A partir del siguiente programa del computador básico con *pipeline*:

```
DATA:  
    res 0  
    i   0  
CODE:  
    MOV B,2  
    mult:  
        MOV A,2  
        ADD A,B  
        MOV (res),A  
        MOV A,(i)  
        ADD A,1  
        MOV (i),A  
        JNE mult  
        ADD A,B  
        MOV (res),A  
        MOV A,(i)
```

Determine el número de ciclos que demora su ejecución detallando en un diagrama los estados del *pipeline* por instrucción. Asuma que el manejo de *stalling* es por *software* a través de la instrucción NOP y que la unidad predictoría de saltos asume que estos nunca se realizan, independiente de que estos sean condicionales o incondicionales. Indique en el diagrama adjunto al enunciado cuando ocurre *forwarding* (con flechas desde el registro hacia la etapa que corresponda), *stalling* y *flushing* (con tachado en las instrucciones *flusheadas*).

Feedback ayudantía

Escanee el QR para entregar feedback sobre la ayudantía.

