DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
ESCUELA DE INGENIERÍA
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

IIC2343 - Arquitectura de Computadores (I/2026)

Ayudantía 15

Ayudantes: Alberto Maturana (alberto.maturana@uc.cl), Mario Rojas (mario.denzel@estudiante.uc.cl), José Mendoza (jfmendoza@uc.cl)

Pregunta 1: Preguntas conceptuales

(a) Indique qué capacidades/instrucciones gana y pierde el computador básico al agregar soporte para paralelismo a nivel de instrucción.

(b) Describa cómo se puede modificar el computador básico con *pipeline* para que las predicciones de salto erróneas pierdan dos ciclos en vez de tres.

(c) Suponga que se modifica la arquitectura del computador básico con pipeline para dividir la etapa MEM en dos nuevas etapas: MEM-R para realizar solo operaciones de lectura; y MEM-W para realiar solo operaciones de escritura. Indique, justificadamente, si esta modificación introduce algún tipo de error **nuevo** necesario de manejar, no es necesario entregar la solución. Puede asumir que la *Jump Unit* y la *Control Hazard Unit* se ubican en la etapa MEM-R.

(d) Si bien la arquitectura del computador básico con *pipeline* es lo suficientemente robusta para resolver un conjunto amplio de *hazards*, existe un caso particular que **no está controlado**.

```
MOV A, 3
MOV B, 0 // Esta instrucción no incide sobre la dependencia de datos
MOV (dir), A // La instrucción previa a la anterior modifica el valor de A
```

Básicamente, no se controla el caso en el que la instrucción previa a la anterior modifica el valor de un registro que **será utilizado para escribir en la memoria de los datos**. A partir de este contexto, conteste de forma justificada las siguientes preguntas:

1. ¿Qué *forwarding unit* debería encargarse de transmitir el valor actualizado de los registros A,B?
2. ¿Qué combinación de señales permite detectar el *hazard* anterior? Indíquelas según el registro intermedio o etapa donde se encuentren y modifique el diagrama del computador básico con *pipeline* para ilustrar tanto la detección como el *forwarding* del valor actualizado. Puede agregar las conexiones y componentes que estime convenientes, pero no debe añadir otra *forwarding unit*.

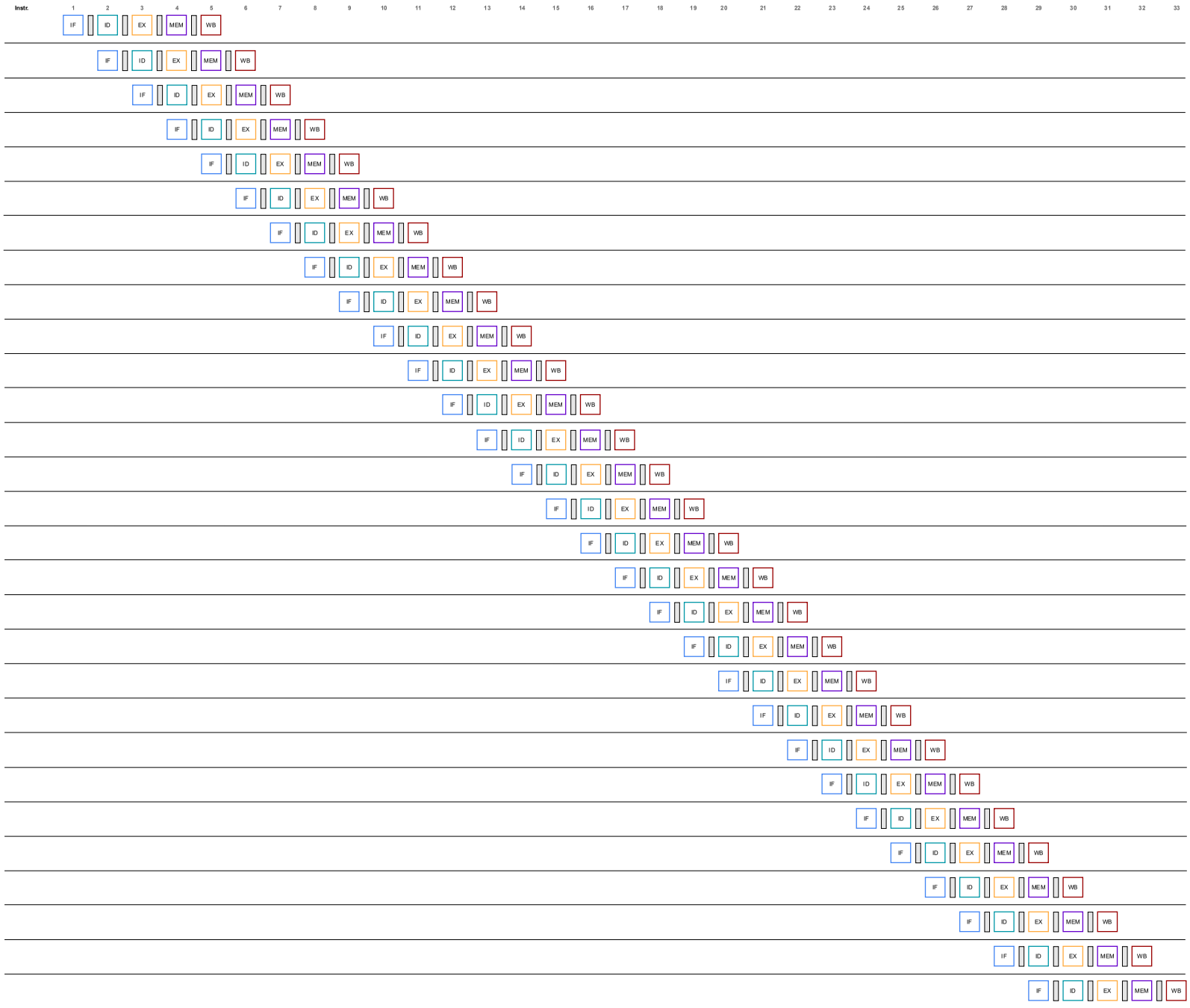
Pregunta 2: Ejecución

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

```
DATA:
  i    0
  arr  11
       13
  len  1
  end  0
CODE:
  MOV A,(i)
  MOV B,(len)
  main:
    JEQ A,B,end // if A == B go to end
    MOV B,arr
    ADD B,A
    MOV A,(B)
    SHL A,A
    MOV (B),A
    MOV A,(i)
    ADD A,1
    MOV B,(len)
    MOV (i),A
    JMP main
  end:
    MOV A,(end)
    ADD A,1
    MOV (end),A
```

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 predictora de saltos asume que estos nunca se realizan, independiente de que estos sean condicionales o incondicionales. Indique en el diagrama cuando ocurre *forwarding* (con flechas desde el registro hacia la etapa que corresponda), *stalling* y *flushing* (con tachado en las instrucciones *flusheadas*). Asuma que las variables del segmento **DATA** se comienzan a almacenar desde la dirección 0x00.

Diagrama vacío



Feedback ayudantía

Escanee el QR para entregar feedback sobre la ayudantía.

