

## Pauta Control 5

Profesor: Yadran Eterovic

### 1. Pregunta 1

Considera el siguiente código escrito para el Assembly de un computador básico de 2 registros con *pipeline*. Indica con un diagrama, si y cuando ocurre *forwarding* (indicando la unidad de *forwarding* correspondiente), *stalling* y *flushing* al ejecutar el código anterior. Considera que el *pipeline* tiene *forwarding* entre todas sus etapas, el manejo de *stallings* es por *software* (instrucciones NOP) y predicción de salto asumiendo que no ocurre.

Finalmente, el programa se ejecuta de la siguiente forma:

```
mov B, (n)      # None
mov A, (i)      # None
nop             # Stalling
add A, 1        # Forwarding Unit 1 mov A, (i)
jne start      # Forwarding Unit 1 add A, 1
mov A, 5        # None
mov (s), A      # Forwarding Unit 2 mov A, 5
mov B, 3        # Flushing mov A, 5 – mov (s), A – mov B, 3
add A, 1        # None
jne start      # Forwarding Unit 1 add A, 1
mov A, 5        # None
mov (s), A      # Forwarding Unit 2 mov A, 5
mov B, 3        # None
```

**Asignacion de puntaje:**

- -0,5 pts por ejecutar un programa distinto.
- -0,25 pts **por instrucción** nombrada incorrectamente o no especificada.
- 0,5 pts FU1.
- 0,5 pts FU2.
- 0,5 pts Flushing.
- 0,25 pts Stalling.
- 0,25 pts FU2 dentro del Flushing

## 2. Pregunta 2

Describe con palabras, de forma clara y precisa, cómo depende el *pipeline* del computador básico de las siguientes unidades de *hardware*: *Forwarding Unit 1*, *Forwarding Unit 2*, *Data Hazard Unit* y *Control Hazard Unit* e indique qué problemas habrían si se eliminara cada una de ellas por separado.

Forward Unit 1:

El pipeline depende de esta unidad para controlar los casos en donde una instrucción previa escribe en registro y la instrucción actual utiliza registro en la ALU.

El problema que podría causar el no tener este componente es que la información del registro estaría desactualizada, por lo que la ejecución de un programa podría ser inconsistente.

Forward Unit 2:

El pipeline depende de esta unidad para controlar los casos en donde una instrucción previa escribe en registro y la instrucción actual escribe en memoria un valor del registro modificado.

El problema que podría causar el no tener este componente es es que la información de la memoria estaría con un dato desactualizado, por lo que la ejecución de un programa podría ser inconsistente.

Data Hazard:

El pipeline depende de esta unidad para controlar los casos en donde la instrucción anterior tiene que estar escribiendo en registro desde memoria y la instrucción actual ocupa como parámetro de la ALU el registro escrito. En estos casos se requiere agregar un mecanismo que permita que la siguiente instrucción espere mientras se obtiene el valor necesario, lo que se conoce como stalling.

El problema que podría causar el no tener este componente sería que el computador no sería capaz de realizar staling en tiempo real, por lo que debería hacerlo a través de software.

Control Hazard:

El pipeline depende de esta unidad para controlar los casos en donde la predicción de salto falla, y envía una señal de flush que avisa a las siguientes etapas del pipeline que no deben ejecutar las señales de control que se habían enviado.

El problema que podría causar el no tener este componente sería que se requeriría de hacer varios stallings para esperar a cada instrucción para saber si hay o no salto.

NOTA: La justificación de los problemas puede variar (como por ejemplo mencionar que siempre habrá inconsistencia). En dicho caso se asigne puntaje en caso intermedio (por no dar razones sobre la instrucción específica).

**Asignacion de puntaje:**

- 0.5 por descripción de como depende el pipeline de cada componente
- 0.25 por describir que problemas habrían si se elimina cada componente