a) Ejecución con "Forwarding" deshabilitado

El programa intercambia el contenido de las palabras de memoria etiquetadas `A` y `B`, y se ejecuta en un simulador con el "Forwarding" deshabilitado.

Instrucciones del programa:

1. `ld r1, A(r0)` — Carga el valor de la dirección `A` en el registro `r1`.
2. `ld r2, B(r0)` — Carga el valor de la dirección `B` en el registro `r2`.
3. `sd r2, A(r0)` — Almacena el valor de `r2` en la dirección `A`.
4. `sd r1, B(r0)` — Almacena el valor de `r1` en la dirección `B`.
5. `halt` — Finaliza la ejecución.

Análisis de los atascos (Stalls) en el Pipeline:

Sin "Forwarding" habilitado, los atascos se deben a las dependencias de datos entre las instrucciones. Específicamente, el problema ocurre debido a las dependencias entre los registros `r1` y `r2` que se leen en las primeras dos instrucciones y luego se escriben en las instrucciones siguientes.

1. **Dependencia de datos entre** `ld r1, A(r0)` **y** `sd r2, A(r0)`:
   - La instrucción `ld r1, A(r0)` carga un valor en `r1` que se usa en la instrucción siguiente (`sd r1, B(r0)`). La instrucción de almacenamiento `sd r1, B(r0)` necesita el valor de `r1` para escribirlo en la dirección de `B`.
   - Sin "Forwarding", el valor de `r1` no estará disponible hasta que se complete completamente la instrucción `ld r1, A(r0)`, lo que genera un **ataque de tipo "Data Hazard"**.
2. **Dependencia de datos entre** `ld r2, B(r0)` **y** `sd r2, A(r0)`:
   - Similar al caso anterior, la instrucción `ld r2, B(r0)` carga un valor en `r2`, y luego la instrucción `sd r2, A(r0)` necesita ese valor para almacenarlo en la dirección de `A`. La instrucción de almacenamiento `sd r2, A(r0)` no puede proceder hasta que `r2` esté completamente cargado.

Tipos de stall:

El tipo de "stall" en este caso es un **"data hazard"** (más específicamente un **"load-use hazard"**), en el que el valor de un registro recién cargado debe ser usado inmediatamente en la siguiente instrucción, pero no está disponible hasta que la instrucción de carga se haya completado.

Promedio de Ciclos por Instrucción (CPI):

Sin "Forwarding", debido a los stalls, el programa probablemente tomará más ciclos por instrucción en promedio. Si asumimos que cada instrucción tarda en promedio 5 ciclos (debido a los stalls de datos), el **CPI promedio** será más alto en comparación con una ejecución sin stalls.

En este caso, podemos estimar el CPI promedio tomando en cuenta que cada instrucción puede tardar más ciclos debido a los atascos:

$$\text{CPI promedio} = \frac{\text{Ciclos totales}}{\text{Número de instrucciones}}$$

En este caso, podría ser aproximadamente **5 ciclos por instrucción**.

b) Ejecución con "Forwarding" habilitado

Cuando el "Forwarding" está habilitado, el procesador puede avanzar los resultados de una instrucción directamente a las siguientes instrucciones que los necesiten, sin esperar a que el valor sea escrito en el registro. Esto elimina los atascos de datos.

Explicación de la mejora:

- El **"Forwarding"** permite que los resultados de las instrucciones anteriores se pasen directamente a las instrucciones que los necesitan. En el caso de las instrucciones de carga `ld`, el valor que se carga puede ser enviado directamente al siguiente paso de la ejecución sin tener que esperar a que la instrucción de carga complete el ciclo de escritura del registro.
- Así, la instrucción `ld r1, A(r0)` puede pasar su valor directamente a la instrucción que lo usa, `sd r1, B(r0)`, sin necesidad de esperar a que se complete la escritura en el registro. Lo mismo sucede con `ld r2, B(r0)` y `sd r2, A(r0)`.

¿Por qué no se presentan atascos?

No se presentan atascos porque el **"Forwarding"** permite que los resultados de la instrucción de carga se pasen directamente al siguiente ciclo de ejecución de la instrucción de almacenamiento, eliminando la necesidad de esperar a que el registro se escriba en memoria.

Color de los registros en la ventana Register:

En muchos simuladores, el color de los registros en la ventana de registros indica el estado del registro. Por ejemplo:

- **Verde o azul**: El valor del registro es válido y está actualizado.
- **Rojo o amarillo**: El registro aún no ha sido actualizado o está en proceso de actualización.

El color de los registros indica cuándo se completan las operaciones y si los valores se están "adelantando" correctamente entre las instrucciones.

Promedio de Ciclos por Instrucción (CPI):

Con el **"Forwarding"** habilitado, los atascos se eliminan, lo que resulta en una ejecución más eficiente. El CPI promedio en este caso debería ser menor. En una arquitectura optimizada con "Forwarding", el CPI promedio puede reducirse a aproximadamente **1 ciclo por instrucción**, ya que la mayoría de las instrucciones pueden ejecutarse sin esperas adicionales debido a la transferencia directa de datos.

Comparación entre ambos casos:

- **Sin Forwarding**: El programa probablemente tiene un **CPI promedio de aproximadamente 5** debido a los stalls por dependencias de datos (atajos de carga).
- **Con Forwarding**: El programa tiene un **CPI promedio de aproximadamente 1**, ya que los resultados de las instrucciones anteriores se adelantan al siguiente ciclo sin necesidad de esperas.

Conclusión:

1. **Sin Forwarding**: Se generan stalls por las dependencias de datos, aumentando el CPI promedio a un valor más alto (aproximadamente 5).
2. **Con Forwarding**: Los atascos se eliminan, lo que resulta en una mejora significativa en el rendimiento, reduciendo el CPI promedio a 1.