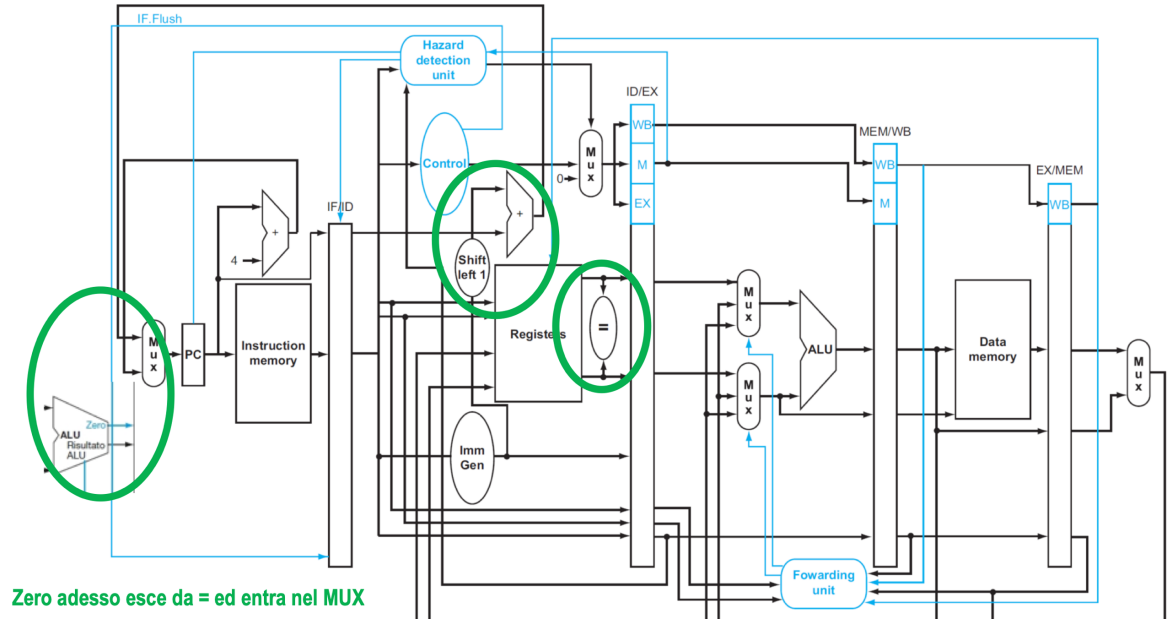
Criticità sul controllo: Branching

- **Ottimizzazione #1: effettuare beq** (e calcolo branch target) nello stadio D



Componenti del Disegno della Pipeline: Hazard Detection Unit, "=" e Forwarding Unit

Hazard Detection Unit (Unità di Rilevamento delle Criticità)
Questa unità ha il compito di rilevare quando si verifica una criticità nella pipeline. Ci sono tre principali tipi di criticità che possono influire sulle prestazioni della pipeline:

1. **Criticità Strutturali**: si verificano quando più istruzioni richiedono la stessa risorsa hardware contemporaneamente.

2. **Criticità sui Dati:** si verificano quando un'istruzione dipende dai risultati di un'istruzione precedente che non è ancora completata.

3. **Criticità sul Controllo:** si verificano principalmente durante l'esecuzione di istruzioni di salto (branch) e riguardano la decisione su quale istruzione eseguire successivamente.

La Hazard Detection Unit è fondamentale per identificare queste situazioni e prendere misure appropriate, come l'inserimento di NOP (no operation) per ritardare l'esecuzione fino a quando la criticità non è risolta.

"=" (Comparatore di Uguaglianza)

Il comparatore di uguaglianza, rappresentato come "=", viene utilizzato principalmente per le istruzioni di salto condizionato (branch). Ad esempio, per l'istruzione `beq` (branch if equal), il comparatore verifica se due registri hanno lo stesso valore. Se sono uguali, la pipeline deve saltare all'indirizzo specificato; se non lo sono, continua con l'istruzione successiva. Nel contesto della pipeline, il risultato del comparatore deve essere disponibile in uno stadio precedente possibile per minimizzare i ritardi. Per esempio, uno schema ottimizzato potrebbe eseguire il confronto durante lo stadio di Decode (ID) invece che nello stadio di Memory (MEM), riducendo il numero di NOP necessari per risolvere il salto.

Forwarding Unit (Unità di Inoltro)
La Forwarding Unit è utilizzata per gestire le criticità sui dati senza dover inserire NOP. Questa unità permette di "inoltrare" i dati appena calcolati da uno stadio della pipeline ad uno stadio precedente che ne ha bisogno. In altre parole, se un'istruzione successiva ha bisogno del risultato di un'istruzione precedente che non ha ancora terminato la sua esecuzione, la Forwarding Unit può prelevare il risultato intermedio e fornirlo all'istruzione successiva, evitando così di dover aspettare che il dato passi attraverso tutti gli stadi della pipeline.

Esempio: se l'istruzione `ADD` produce un risultato che è immediatamente necessario per un'istruzione `SUB`, la Forwarding Unit preleverà il risultato dell'`ADD` dallo stadio di Execution (EX) e lo inoltrerà direttamente allo stadio di Decode (ID) dell'istruzione `SUB`.

Riepilogo del Disegno della Pipeline

1. **Hazard Detection Unit:** rileva quando si verifica una criticità e prende misure per risolverla, come l'inserimento di NOP.
2**. Comparatore di Uguaglianza ("="):** usato nelle istruzioni di salto condizionato per decidere se eseguire un salto.
3. **Forwarding Unit:** gestisce le criticità sui dati inoltrando i risultati intermedi tra gli stadi della pipeline, evitando stalli.

Gestione delle Criticità
Per gestire le criticità nella pipeline, il processore può:
- **Inserire NOP:** per gestire criticità sui dati o sul controllo, introducendo ritardi fino a quando il dato necessario è disponibile.

- Inoltro (Forwarding): per risolvere criticità sui dati senza stalli.

- Predizione dei salti: per minimizzare l'impatto delle istruzioni di salto sul flusso della pipeline.

Ottimizzazioni

1. Calcolo del salto nello stadio ID: per ridurre il numero di cicli persi nelle istruzioni di salto.

2. Branch Delay Slot: eseguire un'istruzione utile immediatamente dopo un'istruzione di salto, che può essere riordinata dal compilatore.

3. Predizione del salto: utilizzo di tecniche di predizione statica o dinamica per minimizzare gli stalli causati dalle istruzioni di salto.

Queste tecniche e componenti sono fondamentali per mantenere l'efficienza e le prestazioni di una pipeline, riducendo al minimo i ritardi e garantendo che le istruzioni vengano eseguite nel modo più rapido possibile.