# Stall Unit

# Overview:

The stall unit in this code is responsible for detecting and handling stall conditions.

A processor pipeline. Stalls occur when there are data threats, especially when a

A read-after-write (RAW) vulnerability has been detected. Stall unit ensures pipeline.

A threat does not progress until it is resolved. Implementation

The stall unit is implemented in the 'Fwd\_Flush\_Stall\_Unit' module. It takes different

Branch prediction (`br\_taken`), current instruction (`inst`), previous information

Read instruction (`inst1`), and enable signals (`rd\_en\_1`). The module provides

Output `stall` and `sel\_rd1`/`sel\_rd2`, which determine whether to stop the pipeline.

And register to read.

## Logic:

The module calculates `raddr1`, `raddr2` and `waddr` based on the instruction fields.

Tests data vulnerability by comparing 'raddr1' and 'raddr2' to 'waddr'. If any danger

Turns out, it sets `stall` to 1 and makes sure both `sel\_rd1` and `sel\_rd2` are 0.

- If no threat is detected, 'stall' is set to 0, and 'sel\_rd1' and 'sel\_rd2' are

`raddr1` and `raddr2` are determined based on the matching `waddr`.

# Flush Unit

# Overview:

The flash unit in this code is responsible for handling branch mispredictions

Pipeline flushing. When an incorrect branch prediction is found, it is rejected

Instructions in the pipeline that should not be executed.

Implementation:

The flush unit is also implemented in the 'Fwd\_Flush\_Stall\_Unit' module. it has

A single output, `flush', which is set to 1 when an incorrect branch guess is detected.

## Logic:

The flash unit checks the `br\_taken` input to determine whether a branch was incorrectly guessed has been created If `br\_taken` is true, indicating a false prediction, it sets `flash` to 1.

Otherwise, 'flash' is set to 0.

These three units work together to ensure proper operation of the pipeline, handling data

Risks, forward data when possible, and respond to branch mispredictions

Flushing the pipeline when necessary.

# Forwarding Unit

# Overview:

The forwarding unit in this code is responsible for forwarding data

The execute stage to the decode stage in the processor pipeline. It helps to solve it

Improve pipeline performance by reducing data risks and stalls.

## Implementation:

The forwarding unit is also implemented in the 'Fwd\_Flush\_Stall\_Unit' module.

It uses the same information and products as the stall unit but provides a different service

Goal: Avoid sending data when possible.

## Logic:

Like the stall unit, the forwarding unit calculates 'raddr1', 'raddr2', and 'waddr'.

Based on the field of instruction.

It checks for data risks, and if no risks are found, it sets 'sel\_rd1' and

`sel\_rd2` allows sending data for up to 1 registers matching `waddr`.

