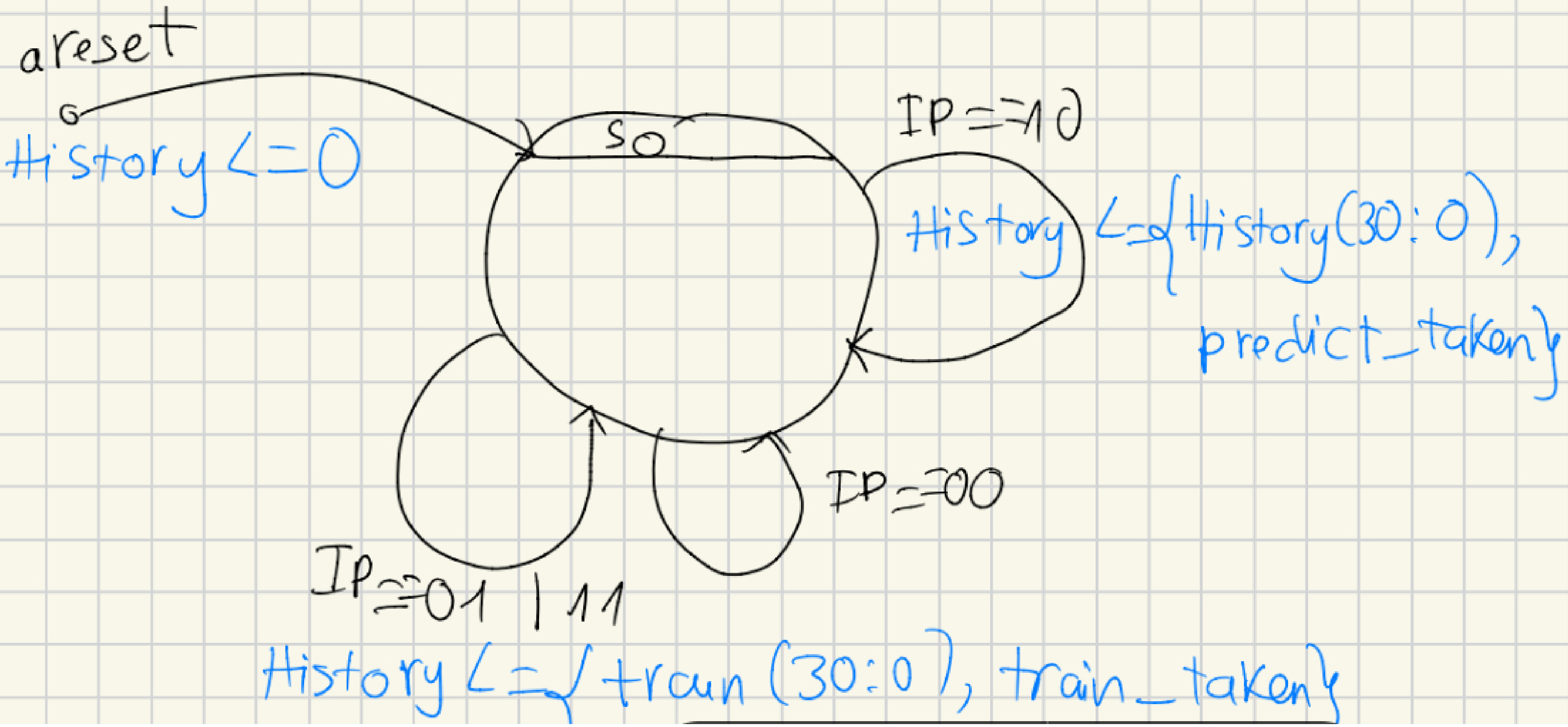
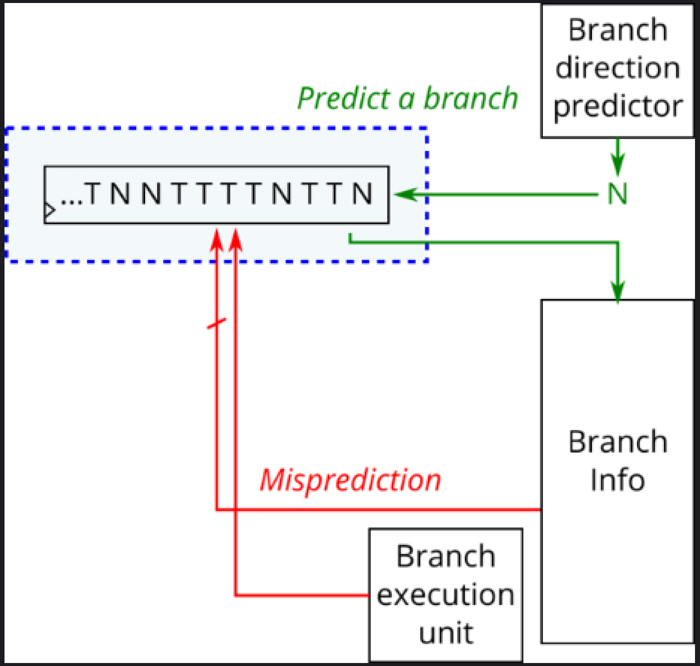
IP = predict_valid, train_mispredicted


areset
History <= 0
S0
IP == 10
History <= {History(30:0), predict_taken}
IP == 00
IP == 01 | 11
History <= {train(30:0), train_taken}

Branch direction predictors are often structured as tables of counters indexed by the program counter and branch history. The branch history is a sequence of "taken" or "not taken" results from recent branches.

In hardware, the branch history register can be implemented as a N-bit shift register. After each conditional branch direction is predicted, its predicted direction is shifted into the shift register. The shift register thus holds the most recent N branch results.



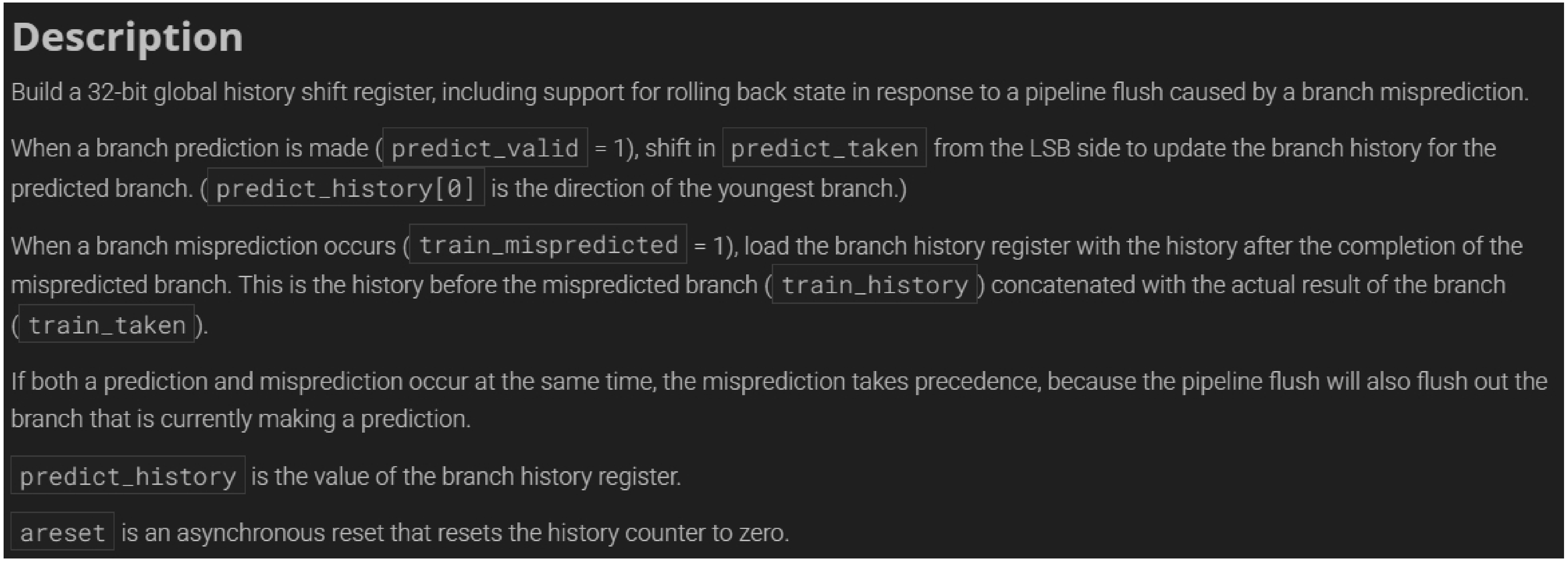
Branch history register with surrounding hardware. This exercise builds the branch history register, inside the blue dashed rectangle.

This diagram shows branch mispredictions being signalled by the branch execution unit, but depending on the processor design, this could also be at retirement or some other point.

Additional complexity arises due to pipeline flushes because branch predictions are done speculatively. When a branch misprediction occurs, the processor state needs to be rolled back to the state immediately *after* the mispredicted branch. This includes rolling back the global history register, which may contain predicted branch results that were shifted-in by branches younger than the mispredicted branch, but now need to be discarded.

We assume here that there is hardware outside of the branch predictor that remembers the state of the branch history register that was used to predict each branch, which is saved for later use for branch predictor training and pipeline flushes. When a branch misprediction occurs, this hardware informs the branch predictor that a branch has mispredicted, the direction the branch should have taken, and the state of the branch history register corresponding to the point in the program immediately *before* the mispredicted branch.

Of course, since the processor restarts to the point *after* the mispredicted branch, the branch history register after the pipeline flush needs to have the actual direction of the mispredicted branch appended.



Description

Build a 32-bit global history shift register, including support for rolling back state in response to a pipeline flush caused by a branch misprediction.

When a branch prediction is made (`predict_valid` = 1), shift in `predict_taken` from the LSB side to update the branch history for the predicted branch. (`predict_history[0]` is the direction of the youngest branch.)

When a branch misprediction occurs (`train_mispredicted` = 1), load the branch history register with the history after the completion of the mispredicted branch. This is the history before the mispredicted branch (`train_history`) concatenated with the actual result of the branch (`train_taken`).

If both a prediction and misprediction occur at the same time, the misprediction takes precedence, because the pipeline flush will also flush out the branch that is currently making a prediction.

`predict_history` is the value of the branch history register.

`areset` is an asynchronous reset that resets the history counter to zero.