CS204
BRANCH PREDICTOR REPORT

objective - objective is to understand working of branch predictor algos and the working of RISCV assembly

To compare the branch prediction algorithms (Always Taken, Always Not Taken, 1-bit Dynamic Branch Predictor, 2-bit Dynamic Branch Predictor), we'll analyze their performance based on accuracy, behavior with different types of code, and theoretical considerations.

1. **Accuracy Comparison:**
Always Taken: Predicts all branches to be taken. It may have high accuracy for linear code sequences but performs poorly for cases with conditional branching.
Always Not Taken:Predicts all branches to be not taken. It may have high accuracy for some linear code sequences but fails for cases with loops or conditional branches.
1-bit Dynamic Branch Predictor:Utilizes a single bit to predict branch outcomes. It may exhibit moderate accuracy but suffers from frequent mispredictions for alternating branches.
2-bit Dynamic Branch Predictor: Utilizes a two-bit saturating counter for predicting branch outcomes. It offers improved accuracy compared to the 1-bit predictor, especially for cases with loops and conditional branches.

2. **Behavior with Different Types of Code:**
Linear Code Sequences: Always Taken and Always Not Taken may perform well for linear code sequences where branches are predictable.
Conditional Branching:Dynamic predictors perform better for cases involving conditional branching, especially when the outcome is data-dependent or varies over time.

Loops: Dynamic predictors with history (2-bit predictor) excel in loop constructs where the branch outcome may change dynamically.

3. **Theoretical Considerations:**
Always Taken/Not Taken: These algorithms are simple and deterministic but lack adaptability to changing branch behaviors.
Dynamic Predictors: These algorithms leverage historical branch behavior to make predictions, allowing them to adapt to changing patterns in code execution.

Testing:

1.Fact_test_Lab_test_Lab
Accuracy(Always Taken): 67.8351 %
Accuracy(Always Not Taken): 32.1649 %
Accuracy(Single bit): 96.704 %
Accuracy(Two bit): 96.3922 %

2.Recursion_test_Lab
Accuracy(Always Taken): 67.8351 %
Accuracy(Always Not Taken): 32.1649 %
Accuracy(Single bit): 96.704 %
Accuracy(Two bit): 96.3922 %

3.qsort_test_Lab
Accuracy(Always Taken): 63.5583 %
Accuracy(Always Not Taken): 36.4417 %
Accuracy(Single bit): 94.7354 %
Accuracy(Two bit): 94.217 %

1.Fact_test_Lab_test_Lab
: "Among the evaluated branch prediction algorithms, the 1-bit Dynamic Branch Predictor exhibited the highest accuracy, achieving 96.704%

accuracy for the Fact_test_Lab_test_Lab case. This suggests that the branch patterns observed within the Fact_test algorithm are conducive to dynamic prediction mechanisms. In contrast, both static predictors (Always Taken and Always Not Taken) yielded lower accuracies, highlighting their limited adaptability to varying branch behaviors. Notably, the 2-bit Dynamic Branch Predictor showcased a significant improvement over the static approaches, achieving an accuracy of 96.3922%. This underscores the effectiveness of leveraging historical branch behavior to enhance prediction accuracy."

2.Recursion_test_Lab
: For the Factorial algorithm, the 1-bit Dynamic Branch Predictor again exhibited the highest accuracy, reaching 94.7354%. This suggests that the Factorial algorithm's branch behavior also benefits from the adaptability provided by dynamic prediction mechanisms. Both static predictors showed notable differences in accuracy compared to Bubble Sort, with the Always Taken predictor achieving 96.704% accuracy and the Always Not Taken predictor achieving 96.3922% accuracy. This indicates the variability of branch behavior across different algorithms.

3.qsort_test_Lab
: In the qsort_test_Lab algorithm, the 1-bit Dynamic Branch Predictor once again demonstrated the highest accuracy, achieving 94.7354%. This suggests that the Binary Search algorithm's branch behavior benefits from the adaptive nature of dynamic predictors. Both static predictors exhibited accuracies closer to each other compared to the previous algorithms, with Always Taken at 63.5583 % and Always Not Taken at 36.4417%. This reflects the inherently more predictable nature of the Binary Search algorithm's branch behavior.

: In the qsort_test_Lab algorithm, the 1-bit Dynamic Branch Predictor once again demonstrated the highest accuracy, achieving 94.7354%. This suggests that the Binary Search algorithm's branch behavior benefits from the adaptive nature of dynamic predictors. Both static predictors exhibited

accuracies closer to each other compared to the previous algorithms, with Always Taken at 63.5583 % and Always Not Taken at 36.4417%. This reflects the inherently more predictable nature of the Binary Search algorithm's branch behavior.

Group members
Wagh tejas 2022csb1144
Priyanshu 2022csb1107
Diveansh verma 2022csb1081

Github
Link-https://github.com/randomtejas345/Assembler-Branch-Prediction-/edit/main/README.md

4. **Conclusion:**
Always Taken/Not Taken: Simple and deterministic, suitable for some cases but limited in adaptability.
Dynamic Predictors: More complex but offer better adaptability and higher accuracy, especially for cases with conditional branching and loops.

5. **Recommendations:**
- For simple and predictable code sequences, Always Taken/Not Taken may suffice.
- For more complex code with conditional branching and loops, dynamic predictors like the 2-bit predictor offer better accuracy and adaptability.