**ÖDEV 2 RAPORU**

A)

Önbellek sisteminin test programı, öncelikle 16384 adet adrese sıralı erişim yaparak değer atar, daha sonrasında ise yine aynı sırayla bu değerleri okur. İlk verilen önbelleksiz sistemde bu işlemin yapılması için her bir erişimde 100 çevrim beklenilmiştir. Tüm yazma ve okuma işlemlerinin tamamlanması yaklaşık 51 milyar ps (pikosaniye) sürmektedir.

Yazdığım önbellek modülü, anabelleğe yapılan erişimlerde en son kullanılan adresleri hafiza\_r adındaki tabloya kaydeder. Bu sayede yakın bir adrese erişilmesi durumunda anabelleğe gitmeye gerek duymadan veri aktarımı gerçekleşir. Belli erişimlerde 100 çevrim bekleme ihtiyacı ortadan kalkmış olur.

Önbellek 128 satırlı ve her satırda 256 bit veri saklıyor. Yani boyutu 4KB. Write-back ve Write-and-Allocate politikalarını uyguluyor. Bunlar sayesinde bir veri değiştirileceği zaman yalnızca önbellekte değiştirilir ve anabelleğe gidilerek 100 çevrim kaybedilmemiş olur. Eğer bir veri yalnızca önbellekte değiştirildiyse o veriye ait kirli bit 1 yapılarak çıkarılması durumunda anabelleğe gidilmesi gerektiği belirtilir. Aynı zamanda anabelleğe bir veri yazılacağı zaman bu veri önbelleğe de getirilir, tekrar erişilmesi durumunda hızlandırılmış olur. Bir satırdaki verinin hangi adrese ait olduğunu anlamak için her satırda 20 bitlik etiket değeri tutulur. Bu etiket değerinin, adresin en anlamlı 20 bitine eşit olması durumunda o satırdaki veri ile eşleşmiş olur.

Yeni önbellek modülüyle tüm yazma-okuma işlemleri yaklaşık 2.8 milyar ps sürmüştür. Kaba bir hesapla tüm sistem yaklaşık 25 kat hızlandırılmıştır. Daha detaylı incelemek gerekirse 32768 adet bellek erişimi yapılmıştır. Bu isteklerin yalnızca 1024 tanesi önbellekte bulunamayıp anabelleğe gitmiştir. 896 tanesi üzerine yeni veri geleceği için önbellekten çıkarılmıştır.

Timeline

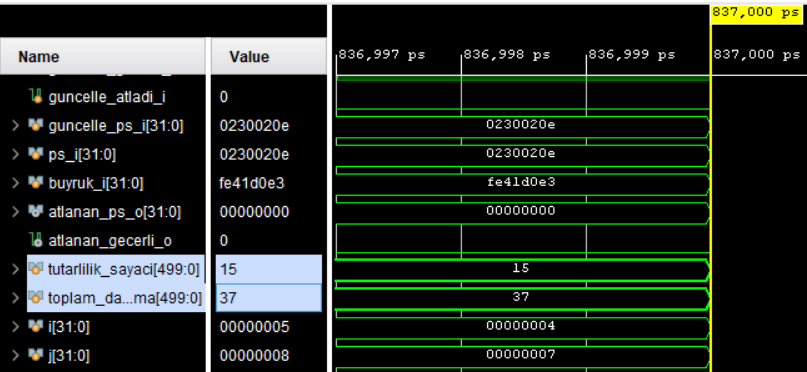
Description automatically generated

Bu veriler ışığında ıska oranı 1024/32768 = 0.03. Yani yaklaşık %3’lük bir ıska oranı var.

B)

Yazdığım öngörü modülü, 4 bitlik genel geçmiş yazmaçlı GShare algoritması kullanıyor. Normal algoritmadan farklı olarak 16 adet 8 satırlı çift doruklu öngörücü tablosu tutuluyor. Normal bir GShare algoritmasına göre daha fazla sayıda farklı dallanma buyruğunu ayrı ayrı hafızada tutabilmesi açısından bu yöntem işe yarayabilir.

Yazdığım bir test koduyla öngörücünün waveformu aşağıdadır.



Bu waveform’a dayanarak 37 adet dallanma buyruğunun 15’inin doğru tahmin edildiği görülmektedir.

Bu doğru tahminler program sayacının olması gerektiği değere daha erken atanmasını sağlayarak işlemcinin başarımını ciddi ölçüde artırabilir. Başarıma tam etkisi pipeline boyutuna ve dallanma buyruklarının çözümlendiği aşamaya bağlıdır.

Hasan TUNA

211101017