

**ÇOKLU İŞ PARÇACIKLARI İLE ÇALIŞMA ZAMANINDA AVF TAHMİNİ
YAPILMASI**

MUSTAFA ÇAVUŞ

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ**

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ARALIK 2014

ANKARA

Fen Bilimleri Enstitü onayı

Prof. Dr. Osman EROĞUL

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

Doç. Dr. Erdoğan DOĞDU

Anabilim Dalı Başkanı

Mustafa Çavuş tarafından hazırlanan ÇOKLU İŞ PARÇACIKLARI İLE ÇALIŞMA ZAMANINDA AVF TAHMİNİ YAPILMASI adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Oğuz ERGİN

Tez Danışmanı

Tez Jüri Üyeleri

Başkan : Yrd. Doç. Dr. A. Murat ÖZBAYOĞLU _____

Üye : Doç. Dr. Ali BOZBEY _____

Üye : Doç. Dr. Oğuz ERGİN _____

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

(İmza)

(Adı Soyadı)

Üniversitesi : TOBB Ekonomi ve Teknoloji Üniversitesi
Enstitüsü : Fen Bilimleri
Anabilim Dalı : Bilgisayar Mühendisliği
Tez Danışmanı : Doç. Dr. Oğuz ERGİN
Tez Türü ve Tarihi : Yüksek Lisans – Aralık 2014

Mustafa ÇAVUŞ

ÇOKLU İŞ PARÇACIKLARI İLE ÇALIŞMA ZAMANINDA AVF TAHMİNİ YAPILMASI

ÖZET

İşlemci tasarımlarının giderek karmaşık hale gelmesi ve uzay boşluğunda radyoaktif maddelerden kopan alfa parçacıklarının yongalara çarpması sistemlerde bit hatalarının oluşmasına neden olabilmektedir. Bazı durumlarda oluşan bu tür hatalar göz ardı edilebilirken, bazı durumlarda sistemin beklenenden farklı bir sonuç vermesine neden olabilmektedir. Bir bit hatasının sistemin çalışmasını ne derecede etkileyeceği, üzerinde çalışan buyruk kümesine ve bit hatasının sistemin hangi bölümünde olduğuna göre değişebilmektedir. Günümüzde bazı işlemcilerde bu tür hatalara karşı koruma sağlayan sistemler bulunmaktadır. Ancak işlemcilerin bu tür hatalara karşı korunması yüksek kaynak ve enerji tüketimine sebep olmaktadır. Bu nedenle buyruk kümelerinin hatalara karşı daha kırılğan olan bölümlerinin belirlenmesi ve bu sayede sadece gerekli görülen zamanlarda sistemin hatalara karşı korunması, çok daha verimli bir kaynak ve enerji kullanımını mümkün kılacaktır. Bu şekilde dinamik olarak oluşabilecek hatalara karşı önlem alınabilmesi sağlanarak işlemcilerin bit hatalarına karşı çok daha verimli bir koruma gerçekleştirilmesi mümkün hale gelecektir. Bu çalışmada, TÜBİTAK'ın 112E004 nolu "Geçici Hatalara Karşı Dayanıklı Mikroişlemciler" projesi kapsamında çoklu iş parçacıkları kullanılarak, buyruk kümelerinin çalışma zamanında hatalara karşı dinamik olarak kırılğanlığını ölçen bir sistem tasarlanmıştır. Buyruk kümelerinin hatalara karşı kırılğanlığının ölçülmesi için AVF (Architectural Vulnerability Factor) tahmini yapılmaktadır. Tasarlanan sistemin geliştirilmesinde M-Sim simülatörü ve ALPHA mimarisi üzerinde çalışılmıştır. Sonuçların analizi SPEC 2006 buyruk kümeleri üzerinde yapılmıştır.

Anahtar Kelimeler: Bit hatası, AVF tahmini, güvenilirlik

University : TOBB Economics and Technology University
Institute : Institute of Natural and Applied Sciences
Science Programme : Computer Engineering
Supervisor : Associate Professor Dr. Oğuz ERGİN
Degree Awarded and Date : M.Sc. – December 2014

Mustafa ÇAVUŞ

ONLINE AVF ESTIMATION USING MULTI PROCESSING

ABSTRACT

As microprocessors scale down and become more complex, the need of considering soft error reliability increases. Soft error reliability makes more sense out of atmosphere because of incearesing reaction of alpha particles. Some processors provides protection against soft error. However, it requires a lot of source and power. In order to provide the required reliability more efficiently, the vulnerability of the instruction sets needs to be estimated online. Online AVF (Architectural Vulnerability Factor) estimation makes it possible to provide more reliable and efficient systems. An Online AVF Estimation method developed in this work using multiprocessing. While the main process is executed as expected, the AVF is estimated using the other copy process in the execution time. In this work, M-Sim simulator is used on ALPHA architecture. Instruction sets from SPEC 2006 is used for benchmarks. This thesis was supported by a TUBITAK project titled “Reliable Microprocessors Design” under grant number 112E004.

Keywords: Soft Error, AVF estimation, reliability

TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren tez danışmanım Doç. Dr. Oęuz ERĐİN'e, projeme destek veren TÜBİTAK'a, yardımlarını esirgemeyen çalıőma arkadaşlarım Serdar Zafer Can ve Emrah İőlek başta olmak üzere bütün Kasırđa Ailesine, çalıőma ortamımızı ve burs olanaklarımızı saęlayan TOBB ETÜ Mühendislik Fakóltesi ve Fen Bilimleri Enstitüsü'ne teőekkürü borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER.....	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
KISALTMALAR.....	xi
SEMBOL LİSTESİ.....	xii
1. GİRİŞ.....	1
1.1. Temel Bilgiler.....	1
1.1.1. Merkezi İşlem Birimi ve Mikroişlemciler.....	1
1.1.2. Soft Error.....	3
1.1.3. Architectural Vulnerability Factor.....	7
1.1.4. M-Sim Simülatör.....	8
1.2. Çalışma Zamanında AVF Tahmini.....	8
2. ÇOKLU İŞ PARÇACIKLARI İLE ÇALIŞMA ZAMANINDA AVF TAHMİNİ YAPILMASI.....	11
2.1. Genel Bakış.....	11
2.2. Yapılan Çalışma.....	12
2.2.1. Temel Algoritma.....	12
2.2.2. İşlem Kopyalama ve Sonlandırma.....	16
2.2.3. Yazmaçlara Bit Hatası Verme.....	18
2.2.4. İşlemler Arası Karşılaştırma.....	20
3. SONUÇLAR VE TARTIŞMA	24

3.1. Sonuların Alınması ve Deęerlendirilmesinde Kullanılan Yöntemler.....	24
3.2. Gerek AVF Deęerinin Hesaplanması.....	25
3.3. Sonuların Analizi.....	26
3.4. Deęerlendirme.....	31
KAYNAKLAR.....	33
ÖZGEMİŐ.....	35

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 1.1. Bazı tam sayıların basit sayma sisteminde (binary) karşılıkları.....	3
Çizelge 3.1. BZIP2, N: 1000, H: 100, Toplam Buyruk Sayısı: 343683, AVF Başına Ortalama Buyruk Sayısı: 7993.....	27
Çizelge 3.2. BZIP2, N: 1000, H: 250, Toplam Buyruk Sayısı: 339713, AVF Başına Ortalama Buyruk Sayısı: 19983.....	28
Çizelge 3.3. BZIP2, N: 1000, H: 500, Toplam Buyruk Sayısı: 359581, AVF Başına Ortalama Buyruk Sayısı: 39953.....	28
Çizelge 3.4. GCC, N: 1000, H: 100, Toplam Buyruk Sayısı: 5984895, AVF Başına Ortalama Buyruk Sayısı: 104998.....	29
Çizelge 3.5. GCC, N: 1000, H: 250, Toplam Buyruk Sayısı: 6030450, AVF Başına Ortalama Buyruk Sayısı: 262193.....	30
Çizelge 3.6. GCC, N: 1000, H: 500, Toplam Buyruk Sayısı: 5777177, AVF Başına Ortalama Buyruk Sayısı: 525198.....	31

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. Intel tarafından üretilen ilk mikroişlemci.....	2
Şekil 1.2. Günümüzden bir mikroişlemci görünümü.....	2
Şekil 1.3. Çekirdekten kopan bir alfa parçacığı.....	5
Şekil 1.4. Uzak boşluğunda yer alan uydular güneş tarafından yapılan alfa ışınlarına maruz kalır.....	6
Şekil 1.5. Bir mikroişlemci yüzeyine çarpan alfa parçacıkları.....	6
Şekil 2.1. Ana işlemin kopyalanması, kopya işleme bit hatası verilmesi, karşılaştırılma yapılması ve kopya işlemin sonlandırılması.....	14
Şekil 2.2 Ana işlemin arka arkaya kopyalanıp hata verilmesi.....	15
Şekil 2.3. Hata verme ve kontrol süreci.....	18
Şekil 2.4. Sonuç yazmacına hata verilme aşaması.....	20
Şekil 2.5. Varsayım için sonuç karşılaştırma aşaması.....	22

KISALTMALAR

Kısaltmalar Açıklama

AVF	Architectural Vulnerability Factor
ACE	Architecturally Correct Execution
FIT	Failures in Time
MTBF	Mean Time Between Failures

SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
<i>N</i>	Bir varsayım sürecinde çalıştırılan, sonuç yazmacı olan buyruk sayısı.
<i>H</i>	Bir AVF tahmini için verilen hatalar ile yapılan toplam varsayım sayısı.
<i>F</i>	Bir AVF tahmini için verilen hatalardan sonucu etkileyeceği varsayılanların sayısı.

İndisler	Açıklama
-----------------	-----------------

Üsler	Açıklama
--------------	-----------------

1. GİRİŞ

1.1. Temel Bilgiler

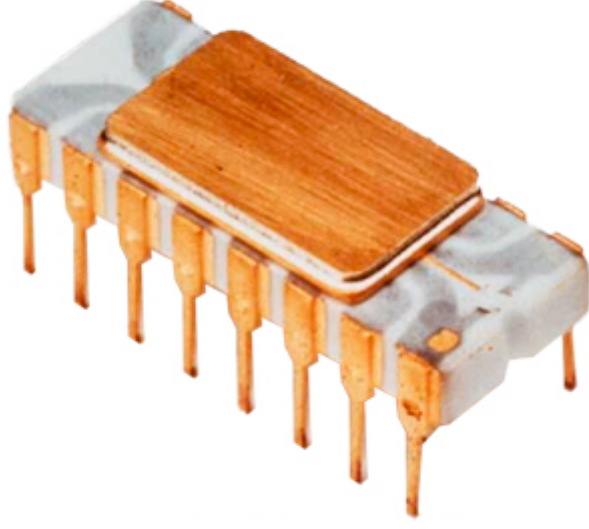
1.1.1. Merkezi İşlem Birimi ve Mikroşlemciler

Merkezi işlem birimi (CPU), bir bilgisayarın beyni ve en temel parçasıdır diyebiliriz. Bir bilgisayarın çalışması sırasında belirli işlemlerin sırayla ve belirli bir düzen içerisinde gerçekleştirilmesi gerekir. Bu işlemlerin düzen içerisinde sırasıyla gerçekleştirilip, gerektiğinde başka birimlerle de veri alışverişi yapılmasını sağlayan eleman merkezi işlem birimidir.

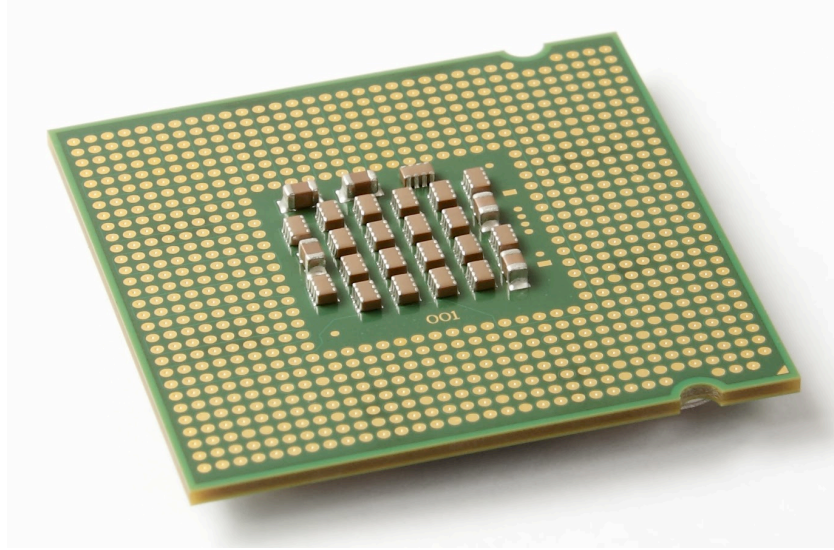
Bilgisayarların ilk zamanların çok büyük bir alan kaplayan merkezi işlem birimleri, 1971 yılında Intel'in binlerce transistörü sadece bir tane silikon üzerinde birleştirmesi ile çok daha küçük bir alan kaplamaya başladı. (Şekil 1.1) Bu sayede merkezi işlem birimleri daha küçük cihazlarda da kullanılabilir ve bilgisayarlar evlere yerleşebilecek kadar küçüldü. Teknolojinin daha da gelişmesi ile merkezi işlem birimleri bugünkü akıllı telefonlar, mp3 oynatıcılar, oyun konsolları gibi birçok ürünün de oluşmasını sağladı.

Mikroşlemciler (Şekil 1.2) günümüzde birçok cihazın içerisinde kullanılan ve tek bir yonga içerisine yerleştirilmiş olan merkezi işlem birimleridir. Mikroşlemcilerin çalışmasını sağlayan elektrik, bu yongaların üzerinde yer alan milyonlarca transistörden geçerek basit matematiksel işlemlerin yapılmasını ve bu şekilde mikroşlemcilerin verilen işlemleri yerine getirmesini sağlar. Basit sayma sistemleri (binary) ile yapılan bu matematiksel işlemler mikroşlemciler üzerinde çok hızlı bir şekilde gerçekleştirilir. Böylece basit sayma sistemine tabi tutularak yapılan mp3 dinleme, video izleme, oyun oynama gibi işlemler mikroşlemciler üzerinde akıcı bir şekilde gerçekleştirilebilir.

Intel 4004



Şekil 1.1. Intel tarafından üretilen ilk mikroişlemci.



Şekil 1.2. Günümüzden bir mikroişlemci görünümü.

1.1.2. Soft Error

Soft error, elektronik sistemlerde ve bilgi işlemede oluşabilen ve bir sinyalin veya verinin bozulması ile ortaya çıkan hatadır. Bu tarz hatalara sistemde yer alan bir kusur neden olabilir.

Soft error 1970'lerden günümüze kadar yaygın olarak bilinen ve dikkate alınan bir konudur. Mikroişlemciler istenen işlemleri gerçekleştirebilmek için gerekli olan verileri basit sayma sisteminde işler ve saklar. Bazı tam sayıların basit sayma sisteminde nasıl ifade edildiği Çizelge 1.1'de gösterilmiştir. Bu verilerin işlenmesi ve saklanması mikroişlemci devrelerinde gerçekleştirilir. Ancak mikroişlemcilerin üretiminde kullanılan materyallerin her zaman dış etkenlere karşı bir duyarlılığı vardır. Mikroişlemciler üzerinde işlenen veriler dışarıdan maruz kalınan etkenler ile değişebilir ve bu da gerçekleştirilen işlemin sonucunu etkileyebilir. Örneğin bir video oynatılırken verilerin beklenmedik bir şekilde dış etkenler tarafından değiştirilmesi görüntünün bozulmasına neden olabilir. Ya da bir robot soft error oluşması nedeni ile kendisinden beklenen görevi doğru şekilde yerine getiremeyebilir. Bu tarz hatalar sisteme bir zarar veremez ancak yapılan işlemin ve hesaplamaların sonucunun hatalı olmasına neden olabilir.

Çizelge 1.1. Bazı tam sayıların 8-bit basit sayma sisteminde (binary) karşılıkları.

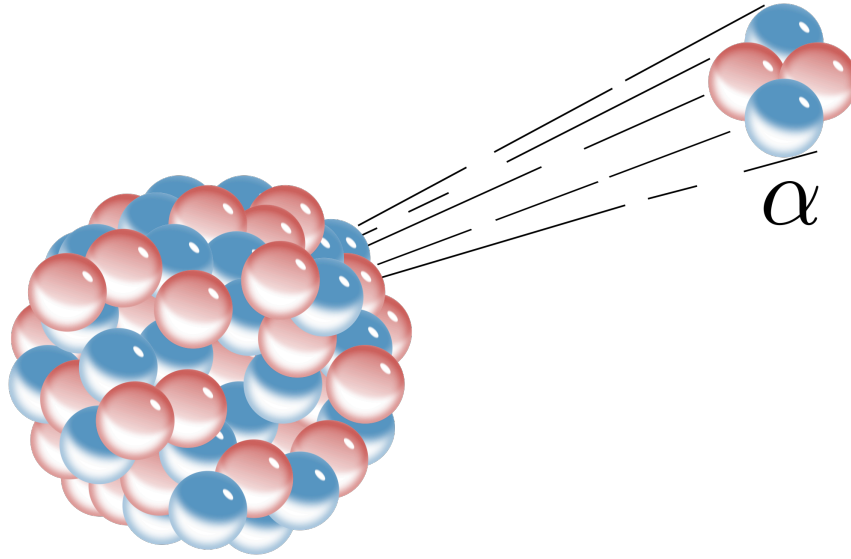
Tam Sayılar	8-bit Binary Karşılıkları
0	00000000
1	00000001
2	00000010
3	00000011
4	00000100
5	00000101
6	00000110
7	00000111
8	00001000
9	00001001

Eğer mikroişlemciler tamamen dış dünyadan izole bir şekilde çalışabilseydi, bu tarz hataların oluşma ihtimalini dikkate almamıza gerek kalmazdı. Ancak uzay boşluğunda oluşan ışımların devre elemanlarını etkilemesi veya devre elemanlarının üretildiği materyallerde oluşan paslanmalar gibi önlenemeyen nedenler soft error oluşmasına neden olabilmektedir ve oluşabilecek bu tarz hataların da dikkate alınması gereklidir.

Soft error iki şekilde oluşabilir. Birincisi yonga seviyesinde oluşan bit hatalarından kaynaklanır. Bu tür hatalar radyoaktif maddelerden çıkan alfa parçacıklarının yonga üzerine çarpması ile tetiklenir. Pozitif yüke ve kinetik enerjiye sahip olan bu alfa parçacıkları yonga içerisinde soft error oluşmasına sebep olabilir. Soft error oluşmasının bir diğer kaynağı ise sistem seviyesinde oluşan bit hatalarıdır. Bu tarz hatalar genellikle veri transferi sırasında bus üzerinde meydana gelir. Veri transferi sırasında oluşan gürültü, verinin hatalı iletilmesine sebep olur ve bu şekilde soft error oluşması tetiklenir.

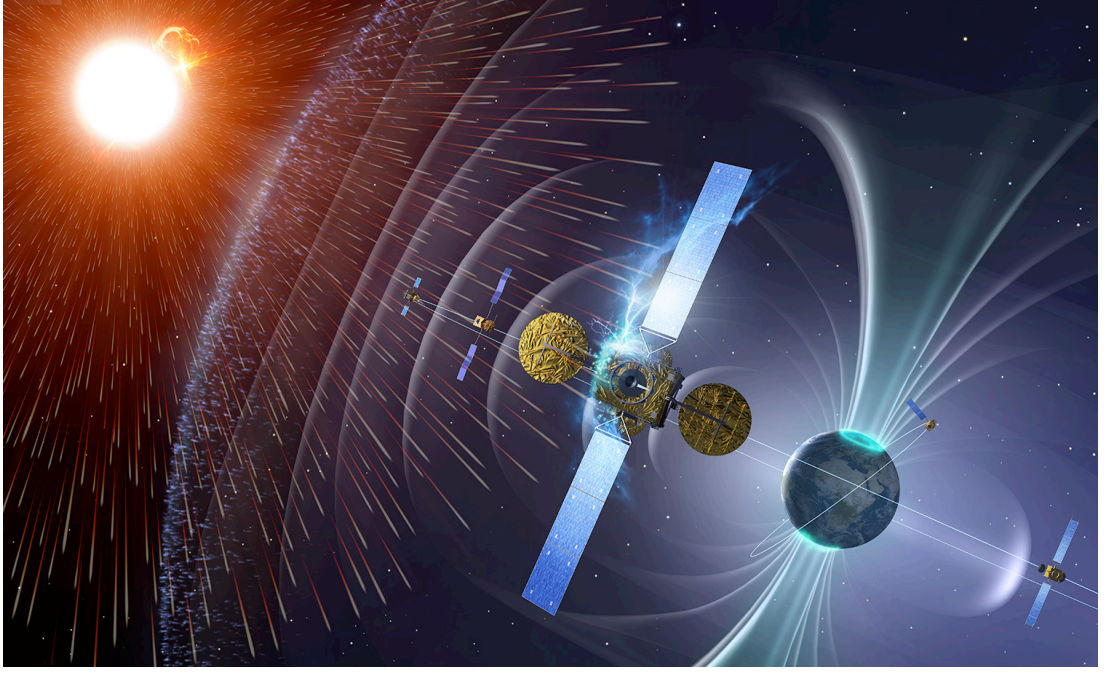
Soft error oluşmasına neden olabilecek dış etmenlerden biri radyoaktif maddelerdir. Uzay boşluğunda radyoaktif çekirdekler tarafından gerçekleştirilen 2 proton ve 2 nötrondan oluşan alfa ışımları yüksek derecede iyonlaştırıcı etkiye sahiptir. (Şekil 1.3) Bu küçük ve yüklü parçacıklar uzay boşluğunda hızla ilerler ve ilk çarptığı materyal üzerinde sahip olduğu yük nedeni ile bir etki yaratır. Genellikle Güneş tarafından yapılan alfa ışımları atmosferde etkisini kaybettiği için bu tür ışımların etkisi en çok uydularda görülmektedir. Bu nedenle uydularda yer alan bilgisayar ve diğer cihazların sahip olduğu mikroişlemciler alfa ışımlarına karşı daha kırılgandır. (Şekil 1.4)

Bir mikroişlemcide alfa ışınlarından kaynaklı bir soft error oluşma ihtimali, birim büyüklükte bir yonga üzerine birim zamanda düşen alfa parçacığı sayısı ile orantılıdır. (Şekil 1.5) Çünkü üzerine daha sık alfa parçacığı çarpan bir yonga üzerinde soft error oluşma ihtimali de artacaktır. Tipik bir ayakkabı tabanının 1 cm²'sine 1 saatte ortalama 0,1 ile 10 arasında alfa parçacığı çarpmaktadır. Genellikle bir sistemin soft error'e karşı güvenli sayılabilmesi için bu oranın 0,001 veya daha az olması istenir.

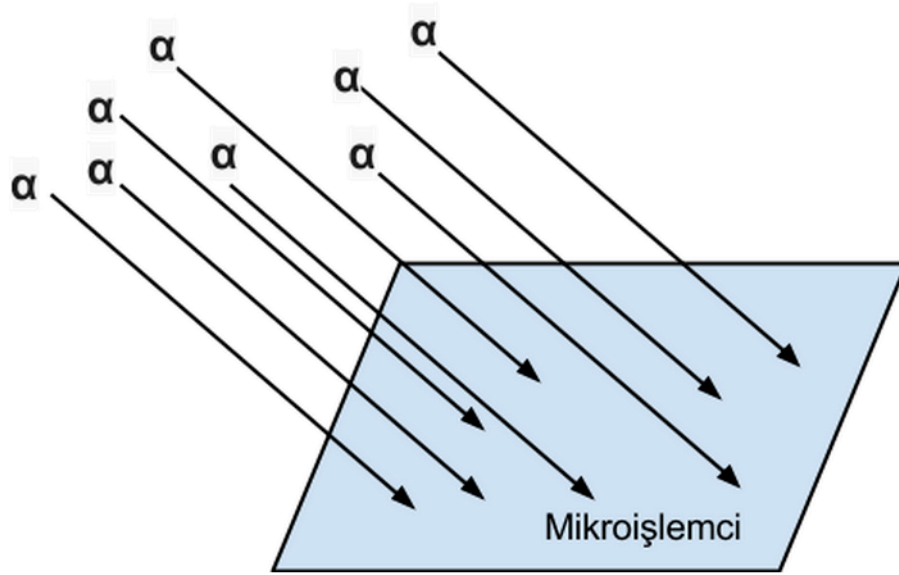


Şekil 1.3. Çekirdekten kopan bir alfa parçacığı

Ayrıca mikroişlemci üzerinde veri transferi sırasında oluşan gürültü de verinin hatalı iletilmesine ve soft error oluşmasına sebep olabilir. Ancak soft error'ün oluşma nedenlerine baktığımızda radyoaktif ışınma gibi diğer nedenlerden dolayı oluşan hataların yanında bu şekilde oluşan hataların oranının önemsizmeyecek kadar az olduğu görülür.



Şekil 1.4. Uzay boşluğunda yer alan uydular güneş tarafından yapılan alfa ışınlarına maruz kalır.



Şekil 1.5. Mikroişlemci üzerine düşen alfa parçacıkları.

Mikroişlemciler tasarlanırken tasarlanan mikroişlemci üzerinde tahmini soft error oluşma sıklığı da hesaplanır. Mikroişlemcinin bu tarz hatalara karşı dayanıklı olması için soft error oluşma sıklığı da düşük olmalıdır. Soft error oluşma sıklığı FIT (failures-in-time) veya MTBF (mean time between failures) ile ölçülür. FIT, mikroişlemcinin 1 milyar saat çalışma süresince yaklaşık kaç tane hata oluşacağı sayısıdır. Yani bir mikroişlemcinin FIT değeri 1 ise, 1 milyar saat çalışan bir mikroişlemcide soft error kaynaklı 1 hata oluşacağı söylenebilir. MTBF ise oluşacak iki soft error kaynaklı hata arasında yıl sayısı olarak geçen süredir. Tasarlanan birçok mikroişlemcinin MTBF değeri o mikroişlemcinin ömründen daha fazladır. Ancak uydularda olduğu gibi atmosfer dışında birim yonga yüzeyi üzerine çarpan alfa parçacığı sayısı da artacağı için soft error oluşma sıklığı da artacaktır. Bu tür durumlarda soft error oluşmasına karşı yeni önlemler almak gereklidir.

1.1.3. Architectural Vulnerability Factor (AVF)

Architectural Vulnerability Factor (AVF), sonraki bölümlerde de sıklıkla bahsedilecek olan bir kavramdır. AVF, sistemin bit hatalarına karşı ne kadar kırılğan olduğunu gösteren bir değerdir. AVF değeri %0 olan bir sistem hatalara karşı tamamen dayanıklıdır. Yani sistemin hangi bitince bir hata oluşursa oluşsun sonuç değişmeyecek ve her zaman doğru olacaktır. AVF değeri 100% olan bir sistem ise hatalara karşı tamamen dayanıksızdır. Yani herhangi bir bitinde hata oluşursa sonuç mutlaka bundan etkilenir ve hatalı olur. Bu çalışmada hesaplanan AVF değerleri 0 ile 1 arasında bir değer olarak ölçüldü. Burada 0, %0'a, 1 ise %100'e karşılık gelmektedir.

Soft error yazmaçlarda, buyruklarda, bellekte veya başka bölümlerde oluşabilir. Bir sistemin farklı bölümlerde oluşan hatalara karşı olan kırılğanlığı da farklı olacaktır. Örneğin bir buyruk kümesi için yazmaçlarda oluşabilecek hatalar için hesaplanan AVF ile buyruklarda oluşabilecek hatalar için hesaplanan AVF değerleri farklı olabilir. Bu çalışmada yazmaçların hatalara karşı olan kırılğanlığı dikkate alınmıştır.

Bir buyruk kümesinin gerçek AVF değerinin hesaplanması buyruk kümesinin bütün buyrukları, kullandığı yazmaçlar ve bellek alanları dikkate alınarak ayrıntılı bir şekilde hesaplanır. (Offline AVF Estimation) Ancak bir buyruk kümesinin gerçek AVF değeri çalışma zamanında hesaplanamayacağı için gerçek AVF değerine yakın tahminler yapabilmemizi sağlayan bazı yöntemler geliştirilmektedir. (Online AVF Estimation) Yapılan çalışmada çalışma zamanında AVF tahmini yapmamızı sağlayacak bir yöntem çoklu iş parçacıkları yardımı ile geliştirilmiştir.

1.1.4. M-Sim Simülatör

M-Sim, çok iş parçacıklı mimariyi destekleyen bir mikroişlemci simülatörüdür. SuperScalar 3.0d tabanlı bir simülatördür ve mevcut sürümü sadece Alpha AXP buyruk kümelerini çalıştırabilmektedir. Superscalar modunda tek iş parçacığı çalıştırabiliyorken, SMT (Simultaneous Multithreaded) modunda birden fazla iş parçacığını aynı anda çalıştırabilir. Tezde anlatılan bütün çalışmalar M-Sim simülatörü üzerinde geliştirilmiş ve test edilmiştir.

1.2. Çalışma Zamanında AVF Tahmini

Soft error nedeni ile oluşan bit hatalarını tespit ederek düzeltebilen ve böylece hatanın sistemin çalışmasını olumsuz etkilemesinin önüne geçilen sistemler mevcuttur. Ancak bu tarz sistemler hatalara karşı koruma sağlarken çok fazla kaynak tüketir ve bu nedenle sistemi yavaşlatır veya enerji tüketimini artırır.

Eğer mikroişlemciler sistemin bit hatalarına karşı on anki kırılgenliğini ölçerek sadece gerektiği anda sistemi korumaya alabilirse, soft error'e karşı korumadan dolayı oluşan aşırı kaynak ve enerji kullanımı da engellenmiş olacaktır. Yani bir buyruk kümesi bazı aşamalarda bit hatalarına karşı daha kırılgen bazı aşamalarda daha dayanıklı olabilir. Eğer buyruk kümesinin hangi aşamalarda hatalara karşı daha kırılgen olduğu bilinirse, sadece gerektiğinde çalışan işlem hatalara karşı korunabilir ve bu da işlemcinin soft error'e karşı çok daha verimli bir koruma sağlayabilmesine imkan verir.

Bir buyruk kümesinin gerçekte hatalara karşı ne kadar kırılğan olduğu hesaplanabilir. Bu o buyruk kümesinin AVF değeridir. Ancak çalışma zamanında buyruk kümesinin o anki gerçek AVF değerini hesaplamak mümkün değildir. Ancak eğer buyruk kümesinin o anki gerçek AVF değerine yakın bir tahmin yapılabilirse, sistemin hatalara karşı doğru zamanda koruma sağlanmasına imkan verilmiş olur.

Bu çalışmada amaç çalışma zamanında belirli aralıklarla gerçeğe yakın bir AVF tahmini yapabilmektir. Böylece çalışma zamanında sistemin bit hatalarına karşı olan kırılğanlığındaki deęişimin ölçülebilmesi sağlanmış olacaktır.

Tahmin edilen AVF değerinin başarısı gerçek AVF değeri ile karşılaştırılarak ölçülür. Ancak burada tahmin edilen AVF değerinin gerçek AVF değerine sayısal olarak yakın olması beklenmemektedir. Burada sadece AVF değerindeki artışların ve düşüşlerin doğru tahmin edilmesi yeterlidir. Yani AVF değerinin buyruk kümesinin belirli bir bölümünde artış gösterdiği ölçülebilirse, sistemin hatalara karşı korunmasının o bölümde aktifleştirilmesi sağlanabilir. Aynı şekilde buyruk kümesinin bir bölümünde AVF değeri düşüş gösteriyorsa, buyruk kümesinin o bölümü çalıştırılırken daha az korumaya ihtiyaç olduğu veya ihtiyaç olmadığı çıkarımı yapılabilir. Bu nedenle tahmin edilen AVF değeri sayısal olarak gerçek AVF değerine yakın olmasa da artış ve düşüşleri gerçek AVF değeri ile paralellik göstermelidir.

Çalışma zamanında AVF tahmini yapabilmek için farklı yöntemler kullanılabilir. Daha önce yapılan birçok benzer çalışmada işlemci mimarisi AVF tahmini yapılabilmesi özel olarak tasarlanmış, bunun için ilgili donanıma ekleme yapılmış ve bunun sonucunda sistemin karmaşıklığı artmıştır. [1-9] Ancak bu çalışmada donanımda AVF tahmini için çok az ekleme yapılarak, büyük ölçüde yazılımsal yöntemlerle AVF tahmini yapılması sağlanmıştır. Burada dikkat edilmesi gereken en önemli şeylerden biri o an çalışmakta olan işlemin AVF tahmini için yapılan işlemlerden etkilenmemesidir. Yapılan çalışmada, çalışma zamanında AVF tahmini yapabilmek için çoklu iş parçacıklarından yararlanıldı. Bu şekilde ana işlem beklenen şekilde çalışmaya devam ederken, ana işlemde kopyalanan bir işleme bit hatası

verilerek buyruk kümesinin hatalara karşı kırılabilirliğini test edilebilmesi sağlandı. Özellikle çok çekirdekli işlemcilerde bu yöntem kullanılarak boşta olan çekirdeklerin AVF tahmini için kullanılması ve böylece yapılan AVF tahmininin ana işlemde herhangi bir yavaşlamaya neden olmaması bir avantaj sağlamaktadır.

2. ÇOKLU İŞ PARÇACIKLARI İLE ÇALIŞMA ZAMANINDA AVF TAHMİNİ YAPILMASI

2.1. Genel Bakış

Online AVF tahmini yapabilmek için uygulama çalıştırıldığı sırada herhangi bir buyruğun sonuç değerinin herhangi bir bitine hata verilmesi ve sonrasında bu hatanın uygulamanın sonucunu etkileyip etkilemediğinin bilinmesi gerekmektedir. Ancak hatanın uygulamanın sonucunu etkileyip etkilemediğini anlayabilmek için uygulamanın beklenen doğru sonucu ile karşılaştırmak gerekir. Bu nedenle aynı uygulama hem tamamen hatasız şekilde çalıştırılmalı, hem de herhangi bir buyruğunun sonuç değerine bir hata verilerek çalıştırılmalı ve bu iki işlem arasındaki fark kontrol edilerek verilen hatanın sonucu etkileyip etkilemediğine karar verilmelidir.

Bunun için önce hatasız çalışacak olan işlem başlatılır. Daha sonra hatasız çalışmakta olan işlem tamamen o anki değerleri ve durumu ile kopyalanır ve birinci işlemin kopyası olan ikinci işlem, birinci işleme paralel olarak başlatılır. Ancak, arada belirli miktar faz farkı oluşmasını sağlamak için, ikinci işlem başlatılmadan önce bir süre bekletilir. Bu şekilde ikinci işleme ait herhangi bir buyruk çalışmasını tamamladığında, aynı buyruğun birinci işlemdeki kopyasının çalışmış ve sonucunun hesaplanmış olduğu garanti altına alınmış olur ve iki işlemin sonuçları arasında karşılaştırma yapmak daha kolay hale gelir.

Kopya işlemin bit hatasından etkilenip etkilenmediğini ölçmek için iki işlemin sonuç değeri olan ve yazmaçlara bu sonuç değerini yazan bütün buyrukları arasında paralel olarak karşılaştırma yapılır. İlk hatanın verildiği sonuç yazmacındaki veriler her zaman farklı olacaktır. Daha sonra devam eden buyrukların sonuç değerlerini yazdığı yazmaçlarda da karşılaştırma yapılmaya devam edilir. Verilen bit hatası devam eden buyrukların sonuçlarını da etkileyebilir. Yani eğer ana işlem ile kopya işlemin buyruklarının yazmaçlara yazdığı sonuç değerleri hata verildikten sonra uzun süre farklılık göstermeye devam ediyorsa, verilen bit hatasının işlemin sonucunu etkileyeceği varsayılabilir. Veya verilen bit hatası devam eden buyruklar tarafından

bir şekilde maskelenerek ortadan kalkabilir. Yani eğer yeterince süre beklenildiğinde iki işlemin buyrukları da yazmaçlara aynı sonuçları yazmaya başlıyorsa, verilen hatanın işlemin sonucunu etkilemeyeceği varsayılabilir.

Ancak AVF tahmini yapabilmek için çalışmakta olan işlemin buyruklarına belirli aralıklarla çok sayıda hata verip her birinin diğerlerinden bağımsız olarak uygulamanın sonucunu etkileyip etkilemeyeceği varsayımında bulunmak gerekir. Bu varsayımlardan alınan veriler yardımı ile AVF tahmini yapabilmek mümkündür.

2.2 Yapılan Çalışma

2.2.1. Temel Algoritma

AVF tahmini yapacağımız buyruk kümesi için başlatılan ana işlem beklenen şekilde çalışmalı ve AVF tahmini sırasında bu ana işlemin çalışması ve sonuçları hiçbir şekilde etkilenmemelidir. Bit hatası sadece ana işlemin kopyası olan diğer işlemin kullandığı yazmaçlara verilecektir. Bu nedenle bir bit hatası verilmeden önce ana işlemin bir kopyası oluşturulmalıdır. Ana işlemin bir kopyası oluşturulduktan sonra kopya işlemin buyrukları tarafından sonuç yazmacı olarak kullanılan bir yazmaca bir bitlik bir hata verilir. Verilen hatanın sonucu etkileyip etkilemeyeceği varsayımını yapabilmek için belirli sayıda buyruğun çalıştırılması beklenir. İstenilen sayıda buyruğun çalışması tamamlandıktan sonra kopya işlem tamamen sonlandırılır. Daha sonra aynı işlemler arka arkaya tekrar edilir ve elde edilen varsayımlara göre ilgili sayaçlar güncellenir.

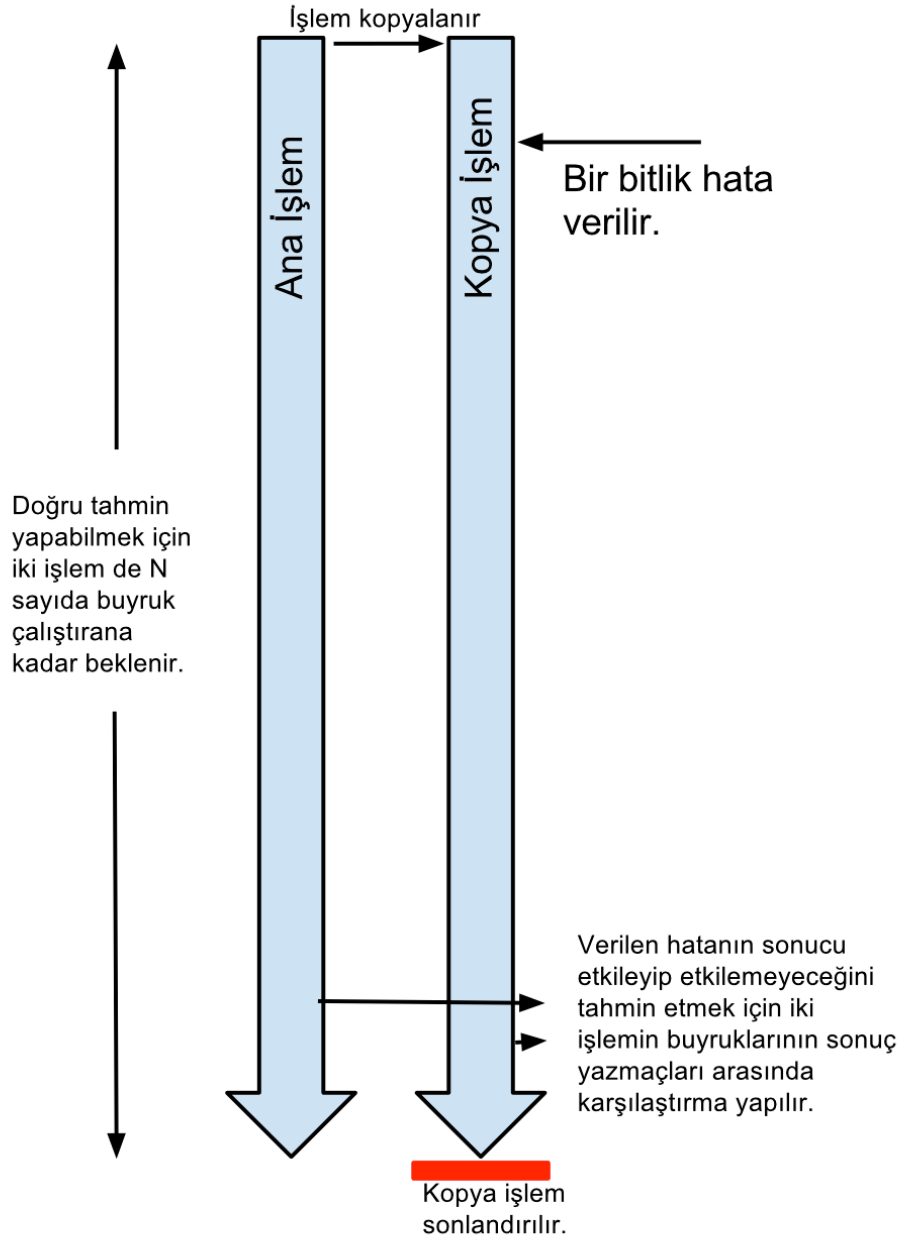
Verilen hatanın N adet sonuç yazmacı olan buyruk çalıştırıldıktan sonra yazmaçlara yazılan değerleri etkileyip etkilemediği kontrol edilerek o hatanın uygulamanın doğru çalışmasını etkileyip etkilemeyeceği varsayımında bulunulabilir. Burada verilen bir hata için yapılan tahmin kesin olmamakla beraber, bir varsayımda bulunabilmek için yeterlidir. Eğer yeterli sayıda hata verilip her biri için bir varsayımda bulunulmuş ise, bu genel olarak verilen hataların işlemin sonucunu etkileme ihtimaline dair gerçeğe yakın bir bilgi verecektir. Bu şekilde işlem çalışır durumdayken gerçeğe yakın bir AVF tahmininde bulunmak mümkündür.

Her N adet sonuç yazmacı olan buyruk için bir sonuç yazmacının bir bitine hata verip bu hatanın sonucu etkileyip etkilemeyeceği varsayımında bulunulursa ve bu işlem arka arkaya yeterli sayıda tekrarlanırsa, gerçek AVF değerine yakın bir tahminde bulunabilmek için yeterli veri elde edilmiş olur. Eğer gerçeğe yakın bir tahminde bulunabilmek için ihtiyaç duyulan varsayım sayısı H ise, $H \times N$ adet sonuç yazmacı olan buyruk çalıştığında gerçeğe en yakın AVF değerini hesaplanmış olur.

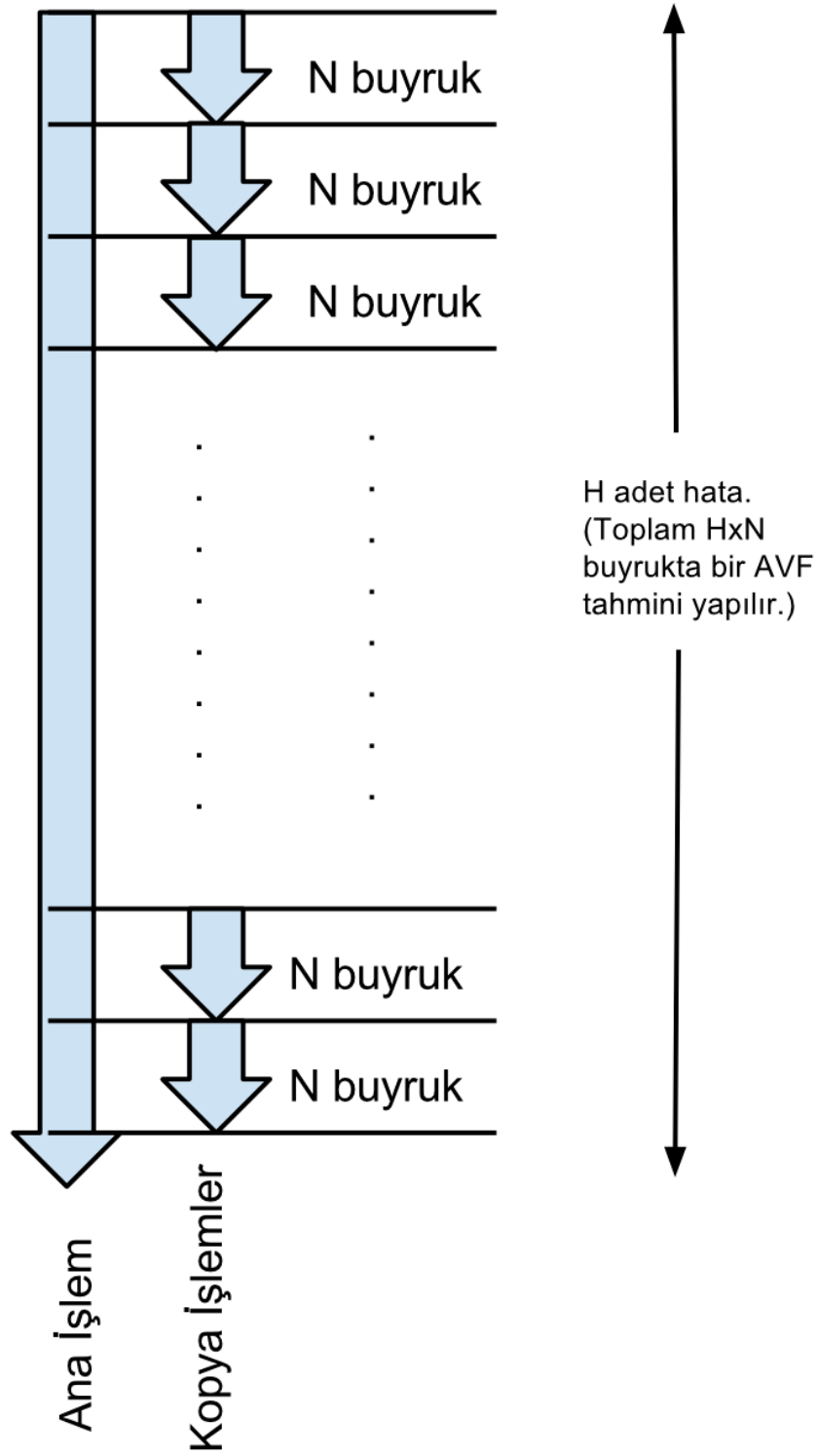
Burada kopya işlemin sonuç yazmaçlarına verilen F adet bit hatasının sonucu etkileyeceği varsayımında bulunulmuş ise ve toplamda H adet bit hatası verilmişse, AVF tahmini için kullanılacak formül F/H olacaktır. Örneğin 100000 buyruk için bir AVF tahmini yapılacak olursa ve her 1000 buyruk için bir hata verip sonucu etkileyip etkilemeyeceği varsayımında bulunulacak olursa toplamda 1000/100000 yani 100 ayrı varsayım yapılmış olur. Bu 100 varsayımdan 25 tanesinin işlemin sonucunu etkileyeceğine karar verilirse, bu 100000 buyruk için yapılacak AVF tahmini 25/100, yani 0.25 olacaktır.

Şekil 2.1, ana işlemin kopyalanması, kopyalanan işlemin bir buyruğunun sonuç yazmacının bir bitine hata verilmesi, N adet buyruk çalıştıktan sonra sonuçların karşılaştırılarak hatanın etkili olup olmayacağı varsayımında bulunulması ve sonrasında kopya işlemin sonlandırılması aşamalarını göstermektedir.

Şekil 2.2, Şekil 2.1'de anlatılmak istenen işlem kopyalama, hata verip varsayımında bulunma ve kopya işlemi öldürme işlemlerinin H adet arka arkaya yapılmasını ve elde edilen varsayımlardan bir AVF tahmini yapılmasını aşamalarını göstermektedir.



Şekil 2.1. Ana işlemin kopyalanması, kopya işleme bit hatası verilmesi, karşılaştırılma yapılması ve kopya işlemin sonlandırılması.



Şekil 2.2. Ana işlemin arka arkaya kopyalanıp hata verilmesi.

2.2.2 İşlem Kopyalama ve Sonlandırma

Eğer buyruk kümesinin hatalara karşı kırılgenliğini test etmek için ana işlemin buyruklarından birinin sonuç yazmacına bit hatası verilirse, hem karşılaştırılabilecek doğru bir sonuç olmadığı için verilen hatanın sonucu etkileyip etkilemeyeceđi çıkarımı yapılamaz, hem de eđer verilen hata işlemin sonucunu etkiliyorsa ana işlemin hatalı çalışmasına neden olunmuş olur. Çalışma zamanında AVF tahmini yaparken ana işlemin işleyişı ve sonuçları bundan etkilenmemelidir.

Ana işlemden bir kopya yaratılır ve paralel bir işlem olarak ana işlemlle beraber çalıştırılırsa, hem verilen hatanın sonucunun ana işlemin sonucunu deđiştirmeyeceđi farklı bir işlem üzerinde çalışma imkanı elde edilmiş olur, hem de verilen hatanın işlem sonucunu etkileyip etkilemeyeceđi varsayımını yapabilmek için karşılaştırma aşamasında sonuçlarının doğruluğundan emin olunan ana işlemin buyrukları referans alınabilir.

Bir önceki kopya işlem görevini bitirip sonlandırıldığında, ana işlemden yeni bir işlem kopyalanır. Ana işlem kopyalandıktan hemen sonra kopya işlem bütün buyruk kuyruđu boşalacak ve hatalı dallanma ihtimallerinden kurtulacak kadar geriye döndürülür. Ana işlemin en son doğru olarak çalıştırılıp yazmaç ve bellek verilerini güncellediđi halinin kopyası tamamen yeni bir paralel işlem olarak en son kaldıđı buyruktan itibaren başlatılır. Bu sırada ana işlemin kontrolündeki yazmaç deđerleri ve bellek verileri de aynı şekilde kopyalanır.

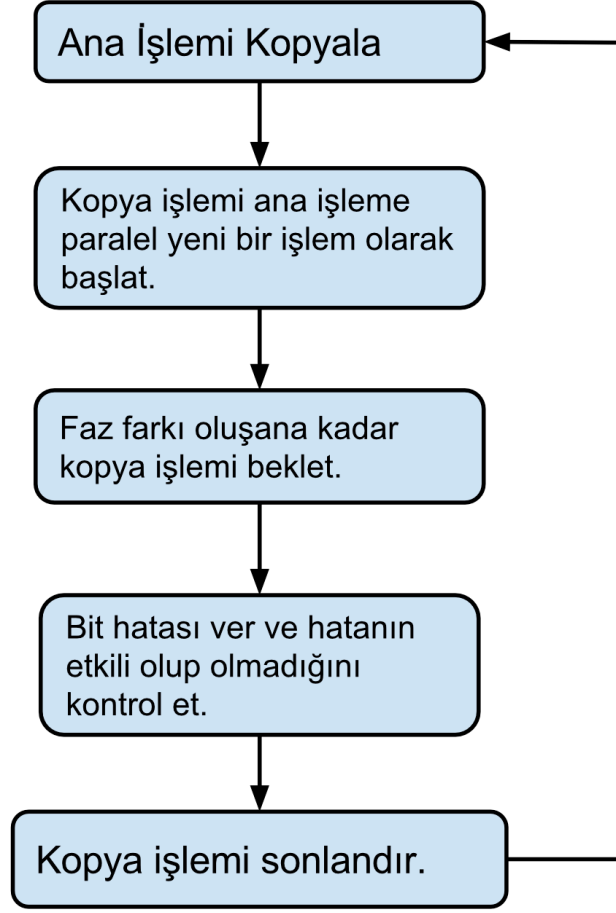
Kopya işlem çalıştırılmadan hemen önce, ana işlem ile kopya işlem arasında bir faz farkının oluşması için ana işlemin belirli sayıda buyruğunun çalıştırılmasının tamamlanması için beklenir. Burada karşılaştırma işleminin kolaylıkla yapılabilmesi için ana işlemin kopya işlemden çok az sayıda buyruk kadar ilerine çalışması yeterlidir. Faz farkı artarsa, bir kopya işlemin sonlandırılıp yeni bir kopya işlemin oluşturulması arasında geçen buyruk sayısı artar ve bu buyruklar AVF hesaplanmasında dahil edilmez. Burada amaç en fazla sayıda buyruđu AVF hesaplamasına dahil etmek olduğundan faz farkı bu çalışmada düşük belirlendi. Ancak faz farkının fazla düşük olması, kopya işlemin zamanla ana işlemin önüne

geçebilme ihtimalini artıracığından ve bu durum her oluştuğunda faz farkını yeniden ayarlamak için kopya işlemin geçici olarak bekletilmesini gerektirdiğinden faz farkını fazla düşük tutmaktan kaçınıldı. Bu çalışmada faz farkı yaklaşık 100 buyruk olarak belirlendiğinde her açıdan en verimli sonuç alınmaktadır.

Faz farkı buyrukların buyruk kuyruğuna aktarılmasından hemen önce ayarlanmaktadır. Kopya işlem başlatıldığında faz farkı istenilen seviyeye gelene kadar kopya işlemin buyruklarının buyruk kuyruğuna aktarılması engellenir. Daha sonra faz farkı istenilen seviyenin altına düşerse, faz farkı tekrar istenilen seviyeye gelene kadar bu işlem tekrarlanır.

Daha sonra bölüm 2.2.3.'te açıklandığı gibi kopya işleme bir bitlik hata verilir ve bölüm 2.2.4.'te açıklandığı gibi N adet buyruk sonra verilen hatanın sonucu etkileyip etkilemediği kontrol edilir. Hatanın sonucu etkileyip etkilemeyeceği varsayımı yapıldıktan sonra kopya işlemin buyruk kuyruğu ve boru hattında yer alan bütün buyruk ve geçici verileri geri alınarak kopya buyruk sonlandırılır. Daha sonra yeni bir bit hatası vermek ve yeni bir varsayım yapabilmek için, ana işlemin yeni bir kopyası oluşturularak bu işlemler tekrarlanır.

Şekil 2.3.'de gösterildiği gibi sırasıyla önce ana işlemin bir kopyası oluşturulur, Ardından kopya işlem ana işleme paralel yeni bir işlem olarak başlatılır. Daha sonra kopya işlem ana işlem ile arasında yeterince faz farkı oluşana kadar bekletilir ve kopya işlemin buyrukları çalışmaya başladıktan sonra kopya işleme bir bit hatası verilip sonucu etkileyip etkilemediği kontrol edilir. Hatanın sonucu etkileyip etkilemediği kontrol edildikten sonra kopya işlemin görevi tamamlanmış olur ve sona erdirilir. Bir AVF tahmini yapabilmek için tekrar bir kopya işlem oluşturularak bu işlemin tamamı arka arkaya tekrarlanır.



Şekil 2.3. Hata verme ve kontrol süreci.

2.2.3. Yazmaçlara Bit Hatası Verme

Buyruk kümesinin hatalara karşı kırılabilirliğini kontrol etmek için ana işlemde bir kopya oluşturulup bu kopya işlemin buyrukları ana işleme paralel olarak çalışmaya başladığında kopya işlemin bir buyruğunun sonuç yazmacına bir bitlik bir hata verilir. Burada hangi buyruğun sonuç yazmacına hata verileceği ve o yazmacın hangi bitine hata verileceği belirlenmelidir.

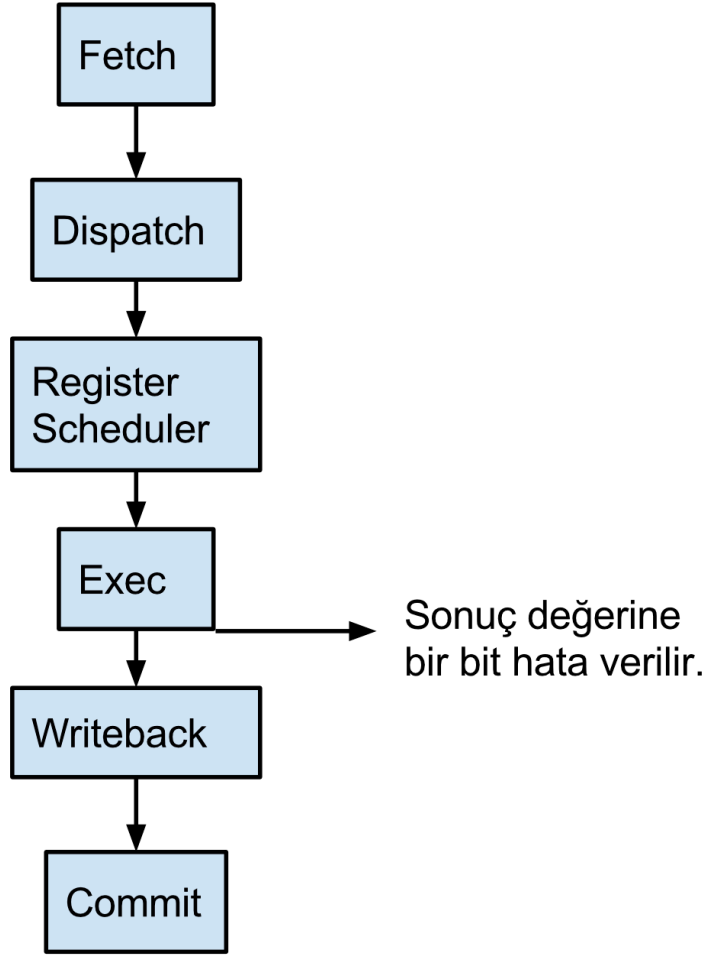
Bir bit hatası verebilmek için, öncelikle sonuç yazmacı olan bir buyruk seçilmelidir. Bit hatası vermek için kopya işlemin çalışmaya başladıktan sonraki birkaç buyruğundan bir rastgele seçilebilir. Yapılan testlerde kopya işlem çalışmaya başladıktan sonuç yazmacı olan ilk buyruk seçildi. Eğer daha sonraki buyruklardan

birini seçersek, daha fazla buyruğu AVF hesaplamasının dışında tutmuş oluruz. Burada sürekli ilk buyruğun seçilmesi AVF tahmininin başarısı açısından önemli bir fark yaratmayacağı için bit hataları her zaman kopyalanan işlemin sıradaki ilk buyruğunun sonuç yazmacına verilmiştir.

Hatanın ilgili yazmacın hangi bitine verileceği ise yapılacak AVF tahmininin gerçeğe yakın olması için önemlidir. Bu nedenle yazmacın hangi bitine hata verileceği tamamen rastgele seçilir. Daha sonra rastgele seçilen bitin 1 ile XOR işleminin verdiği sonuç, yeni hatalı bit olacaktır.

Bu hatanın boru hattının hangi aşamasında verileceği de AVF tahmininin başarısı için önemlidir. Eğer hata buyruk sonucu hesaplamadan verilirse, daha sonra hesaplanan sonuç hatayı maskeler ve verilen hataların etkisi görülmez. Eğer hata çok geç verilirse, hatalı dallanma ile atılacak buyruklar veya boru hattında sonuçları sonradan başka buyruklar tarafından maskelenen buyruklar göz ardı edilmiş olabilir. Ancak gerçeğe yakın bir AVF tahmini yapabilmek için, hatanın bu durumlarda maskelenebileceğini de dikkate almak gereklidir.

Bir buyruğun sonuç değerine hata vermek için en uygun yer boru hattının hesaplama aşamasıdır. Bu aşamada buyruğun sonucu hesaplandığında elde edilen sonuç değerinin rastgele bir biti seçilir ve bu bit 1 ile XOR işlemine sokularak bir hata oluşmasını taklit edecek şekilde değiştirilir.



Şekil 2.4. Sonuç yazmacına hata verilme aşaması.

2.2.4. İşlemler Arası Karşılaştırma

Kopya işlemin bir buyruğunun sonuç yazmacının bir bitine hata verildiğinde hatalı bit sonradan diğer buyruklar tarafından maskelenebilir veya diğer buyruklar tarafından taşınarak işlemin sonucunu etkileyebilir.

Kopya işlemin bir buyruğunun sonuç yazmacına verilen bit hatasının işlem sonucunun etkileyip etkilemeyeceği varsayımında bulunabilmek için hatanın verildiği buyruktan itibaren N adet sonuç yazmacı olan buyruk çalışana kadar beklenir ve bu buyrukların en son çalıştırılan bir bloğunun sonuç yazmaçlarına yazdıkları değerlerde hata olup olmadığı kontrol edilir. Eğer verilen bit hatası sonraki buyruklar ile taşınarak kontrol edilen blokta yer alan buyrukların sonuçlarını

etkilemeye devam etmiş ise, verilen bit hatasının işlemin sonucunu etkileyeceği varsayılabilir. Eğer verilen bit hatası kontrol edilen bloğa kadar maskelenmiş ve ortadan kalkmış ise kontrol edilen bloğun buyruklarının sonuç yazmaçları hatadan etkilenmez ve doğru değerleri gösterir.

Hatanın sonucu etkileyeceği varsayımlarının sayısı F olsun. Toplamda H defa hata verip bir varsayımda bulunduğumuzu düşünelim. Bu durumda tahmin edeceğimiz AVF değeri aşağıdaki gibi hesaplanır. (Denklem 2.1)

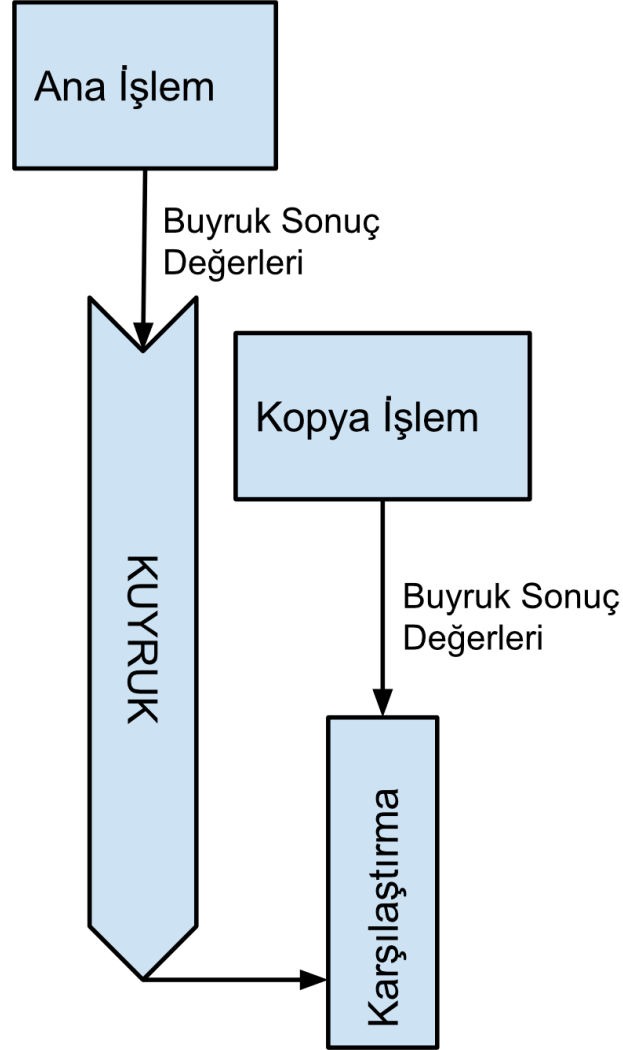
$$AVF_{tahmin} = \frac{F}{H} \quad (2.1)$$

Yeni bir AVF tahmini yapılacağı zaman hem F hem de H değeri 0'lanır. Bu andan itibaren her verilen hata için karşılaştırma sonuçlandığında H değeri bir artırılır. Eğer yapılan karşılaştırma sonucunda verilen bit hatasının işlemin sonucunu etkileyeceği varsayımı yapılmışsa F değeri bir artırılır. Örnek olarak toplamda 100 defa hata verilir bu hatalar için yapılan karşılaştırmalarda 25 tanesinde verilen hatanın işlemin sonucunu etkileyeceği varsayımı yapılırsa tahmin edilen AVF değeri 0.25 olacaktır.

Sonuçların karşılaştırılması boru hattının işleme (commit) aşamasında gerçekleştirilir. Ana işlemin buyruklarının sonuçlarını karşılaştırmak üzere kayıt altına almak için bir kuyruk veri yapısı (queue) kullanılmaktadır. Ana işlemin buyruklarının sonuçlarının tutulacağı kuyruk, yeni bir AVF tahmini yapılmadan hemen önce tamamen boşaltılır. Kopya işlem oluşturulduktan sonra ilk buyruktan itibaren sonuç yazmacı olan bütün buyrukların sonuçları boru hattının işleme aşamasında kuyruğa eklenir. Kopya işlem faz farkı oluşturmak için bekletilirken, kuyruk da faz farkı oluşana kadar ana işlemin buyruklarının sonuç değerleri ile dolacaktır.

Kopya işlemin buyrukları boru hattının işleme aşamasına gelmeye başladığı andan itibaren eğer sonuç yazmaçları varsa, bu yazmaçların değerlerini kuyruğun en başındaki değer ile karşılaştırır. Karşılaştırmadan sonra kuyruğun en başındaki değer kuyruktan çıkarılır. Eğer karşılaştırma hatalı ise, o buyruğun verilen bit hatasından

etkilendiđi sylenebilir. Karşılařtırma hatalı deđil ise o buyruk verilen bit hatasından etkilenmemiřtir. Eđer diđer buyrukların sonuđları aynı řekilde dođru karşılařtırılabiliyorsa, verilen bit hatasının iřlemin sonucunu etkilemeyeceđi varsayılabılır. Aksi durumda verilen bit hatasının sonucu etkilediđini varsayılır ve F deđer bir artırılır.



řekil 2.5. Varsayım için sonuđ karşılařtırma ařaması.

Burada ana işlem ile kopya işlem arasındaki faz farkının korunması önemlidir. Ana işlem buyrukları her zaman kopya işlemin aynı buyruklarından önce çalışmalıdır. Aksi durumda kuyruktaki veri sayısı 0'a düşecektir ve kopya işlemin sonuç yazmacı olan bir buyruğu boru hattının işleme aşamasına geldiğinde o buyruğun sonucunu karşılaştıracak bir veri bulunamayacaktır. Çünkü ana işlemin o veriyi hesaplayacak olan buyruğu henüz çalışmamış olacaktır. Bu nedenle faz farkını korumak ve AVF tahmini yapılana kadar kuyruğu dolu tutmak gerekmektedir. Eğer kopya işlem tarafından kuyrukta yer alan bir veri çekilecekse, o veri ana işlem tarafından daha önceden kuyruğa mutlaka eklenmiş olmalıdır. Karşılaştırma aşamaları ve kuyruğun kullanımı Şekil 2.5'te gösterilmiştir.

3. SONUÇLAR VE TARTIŞMA

3.1. Sonuçların Alınması ve Değerlendirilmesinde Kullanılan Yöntemler

Yapılan AVF tahmini yöntem olarak gerçek AVF ölçümünden farklı olduğu için tahmin edilen AVF ile gerçek AVF sayısal değerlerinin birbirine yakın olması beklenmemektedir. Yapacağımız AVF tahmininin amacı, AVF değerindeki artışları ve düşüşleri algılayıp ona göre sistemin önlem alabilmesine imkan vermektir. Yani gerçek AVF artış gösterdiğinde tahmin edilen AVF de artış gösterirse veya tam tersi olacak şekilde gerçek AVF düşüş gösterdiğinde tahmin edilen AVF de paralel olarak düşüş gösteriyorsa yapılan tahminlerin gerçekçi olduğu iddia edilebilir.

Sonuçlar ALPHA mimarisinde M-Sim simülatörü üzerinde alınmıştır. Sonuçların alınmasında SPEC 2006 buyruk kümeleri kullanılmıştır.

Sonuçlarda belirtilen N değeri, bir varsayım için kaç tane sonuç yazmacı olan buyruk çalıştırıldığı bilgisidir. Her varsayım için çalıştırılan toplam buyruk sayısı daha fazladır. N değeri çok düşük olduğunda verilen hatanın maskelenmesi veya etkisiz olduğu sonucunun anlaşılması ihtimali azalır. Bu da tahmin edilen AVF değerinin olması gerekenin çok üzerinde hesaplanmasına neden olur. N değerinin çok yüksek verilmesi hatanın sonucu etkileyip etkilemeyeceği konusunda daha gerçekçi bir varsayım yapmamızı sağlasa da, aynı aralıkta AVF tahmini yapmak istediğimizde bir AVF tahmini yapılması için kullanılan varsayım sayısının (H) azalmasına neden olur ve bu da tahmin edilen AVF değerinin başarısını olumsuz etkiler. Sonuçların Analizi (3.3) bölümünde, her buyruk kümesi için farklı N değerleri ile sonuç alınmıştır.

Sonuçlarda belirtilen H değeri bir AVF tahmini yapabilmek için kaç adet hata verilip varsayım yapıldığını göstermektedir. H değeri fazla düşük verildiğinde daha az varsayım kullanılarak AVF tahmini yapılacağı için yapılan tahminin başarısı düşecektir. F değeri çok yüksek verildiğinde yapılan AVF tahmininin başarısı artacaktır ancak bir AVF tahmini için çok fazla buyruğun çalışmasını beklemek gerekecektir. Bu nedenle burada H değeri başarılı bir AVF tahmini yapılabilen en düşük değer olmalıdır. Sonuçların Analizi (3.3) bölümünde her buyruk kümesi için farklı H değerleri ile sonuç alınmıştır.

Her AVF tahmini yapıldığında ana işlemin kaçınıcı buyrukta olduğu Sonuçların Analizi (3.3) bölümünde çizelgelerin açıklamaları üzerinde gösterilmiştir. Yapılan AVF tahminleri ve karşılaştırıldığı gerçek AVF değerleri, bir önceki AVF değerinden itibaren hesaplanmıştır. Her AVF tahmini ve gerçek AVF değerinin hesaplanmasının ardından yeni bir tahmin yapabilmek için o an çalışmakta olan buyruktan başlayarak en baştan yeni bir AVF tahmini ve gerçek AVF hesaplaması yapılır.

3.2. Gerçek AVF Değerinin Hesaplanması

Tahmin edilen AVF değerlerinin başarısını ölçmek için gerçek AVF değerlerini referans almak gerekir. Gerçek AVF değerlerini hesaplamak için kullanılacak yöntemlerden biri Little's Law [10]'dur. Buna göre aşağıdaki denklem bize gerçek AVF değerini verecektir. (Denklem 3.1)

$$AVF_{gerçek} = \frac{B_{ace} \times L_{ace}}{B_{toplaml} \times L_{toplaml}} \quad (3.1)$$

Burada B_{ace} yazmaçlardaki toplam ACE (Architecturally Correct Execution) [11] bit sayısıdır. ACE bitler, bir hata oluştuğunda sonucu etkileyecek olan bitlerdir. Bunun haricindeki, yani bir hata oluştuğunda sonucu etkilemeyecek olan bitler ise un-ACE [11] bitlerdir. L_{ace} ise, ACE bitlerin varlığını sürdürdüğü toplam çevrim sayısıdır. $B_{toplaml}$ yazmaçlardaki toplam bit sayısı, $L_{toplaml}$ da toplam çevrim sayısıdır.

Başlangıçta yazmaçlarda yer alan bütün bitlerin ACE bit olduğu kabul edilir. Daha sonra un-ACE olan bitler bulunur. Diğer bitler ACE bit olarak kalacaktır. Ayrıca her yazmaç için ACE bitlerin ömrünün kaç çevrim olduğu da ölçülür.

Bir yazmaca bir buyruk tarafından bir değer yazıldığında o yazmaç takip edilir. Eğer yazmaca yazılan değer başka bir buyruk tarafından okunmadan tekrar o yazmaca bir değer yazılıyorsa o yazmaca ait bitler un-ACE olarak kabul edilir. Yani o bitlerde oluşacak herhangi bir hata sonucu etkilemeyecektir. Aksi durumda o yazmaca ait olan bitler ACE olarak kabul edilir.

3.3. Sonuçların Analizi

SPEC 2006 buyruk kümeleri üzerinde yapılan AVF tahminlerinde gerçek AVF değeri ile istenildiği kadar yakınlık olmasa da paralel artışlar ve düşüşler yakalanabilmiştir. Ancak H değeri düşük tutulduğunda tahmin edilen AVF değerinde gerçeğe göre daha fazla sapma görülmektedir.

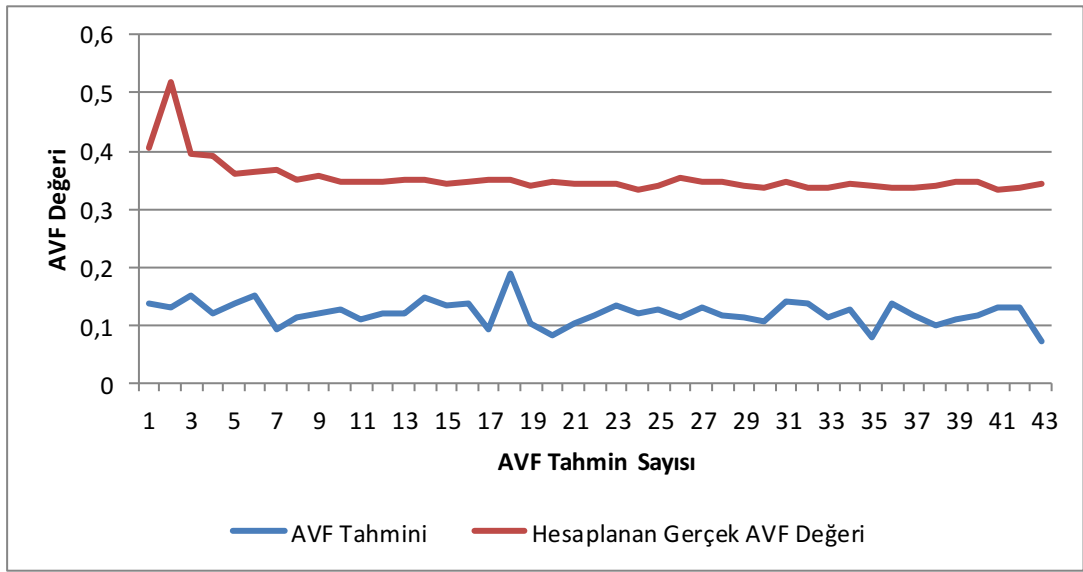
Sonuçlar iki farklı buyruk kümesi ile ve her birinde 3 farklı H değeri ile alınmıştır. N değerinin 1000'in üzerinde olması yapılan tahminin başarısını olumlu etkilemediği için ve aksine bir AVF tahmini yapılabilmesi için daha fazla buyruğun çalışmasını beklemeye neden olduğu için alınan sonuçlarda N değeri 1000 olarak verilmiştir.

Sonuçlar BZIP2 ve GCC buyruk kümeleri ile alınmıştır. İkisi için de sırasıyla 100, 250 ve 500 H değerleri ile sonuç alınmıştır.

Çizelge 3.1'de, BZIP2 buyruk kümesi ile ve H değeri 100 olduğunda alınan sonuç gösterilmektedir. Çizelgede görüldüğü gibi düşük H değeri için tahmin edilen AVF değerinde gözlemlenen değerlerde değişim çok daha belirgindir. AVF tahminlerinin genel seyri aşırı bir tutarsızlık göstermese de, çizelgeden de görülebildiği gibi AVF tahmin aralıkları kısa tutmak istenilen sonucu vermeyecektir. Bu durumda sağlıklı bir AVF tahmini için H değerinin daha yüksek olması gerekmektedir.

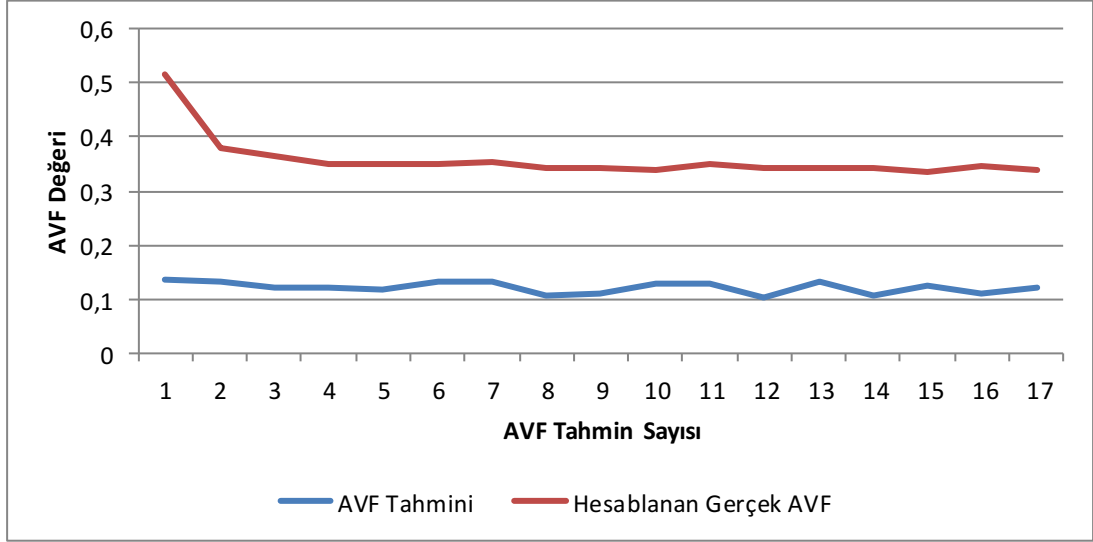
Çizelge 3.2’de BZIP2 buyruk kümesi daha yüksek bir H değeri ile çalıştırılmıştır. Çizelgede görüldüğü gibi H değeri 250 olduğunda AVF tahmini gerçeğe biraz daha yakın ilerleyiş göstermektedir. AVF tahmininin genel ilerleyişi gerçeğe bir nebze yakındır ancak bazı noktalarda, özellikle başlangıç ve bitişlerde istenilen seviyede değildir.

Çizelge 3.1. BZIP2, N: 1000, H: 100, Toplam Buyruk Sayısı: 343683,
AVF Başına Ortalama Buyruk Sayısı: 7993

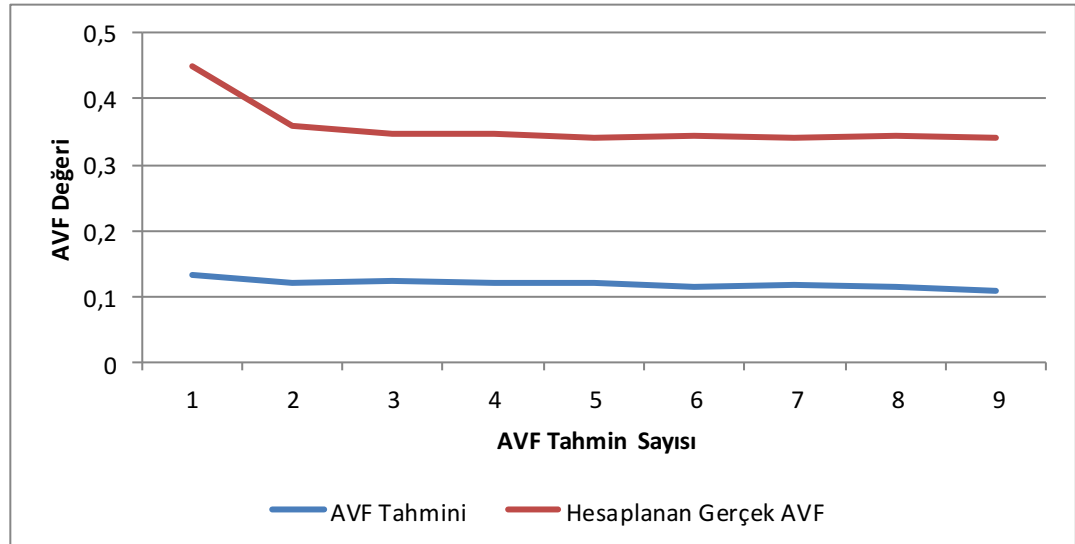


Çizelge 3.3’te BZIP 2 buyruk kümesi 500 H değeri ile çalıştırılmıştır. BZIP2 buyruk kümesinin AVF değerleri için değişimi gözlemlemek zor olsa da yapılan tahminlerin genellikle tutarlı olduğunu söyleyebiliriz. Ancak bazı noktalarda yapılan tahminler yine de istenilen seviyede değildir. Örneğin Çizelge 3.3’te görüldüğü gibi gerçek AVF değeri yüksek bir değerden başlayıp hızla düşüşe geçiyorken tahmin edilen AVF değerinde bu düşüş çok daha az ve önemsiz görünmektedir. Ancak ilerleyiş hem gerçek hem de tahmin edilen AVF değerleri için çok küçük değişimler ile yavaş yavaş azalarak devam etme eğilimi gösterdiği için sonucun genel olarak tutarlı olduğunu söylemek mümkündür.

Çizelge 3.2. BZIP2, N: 1000, H: 250, Toplam Buyruk Sayısı: 339713,
AVF Başına Ortalama Buyruk Sayısı: 19983

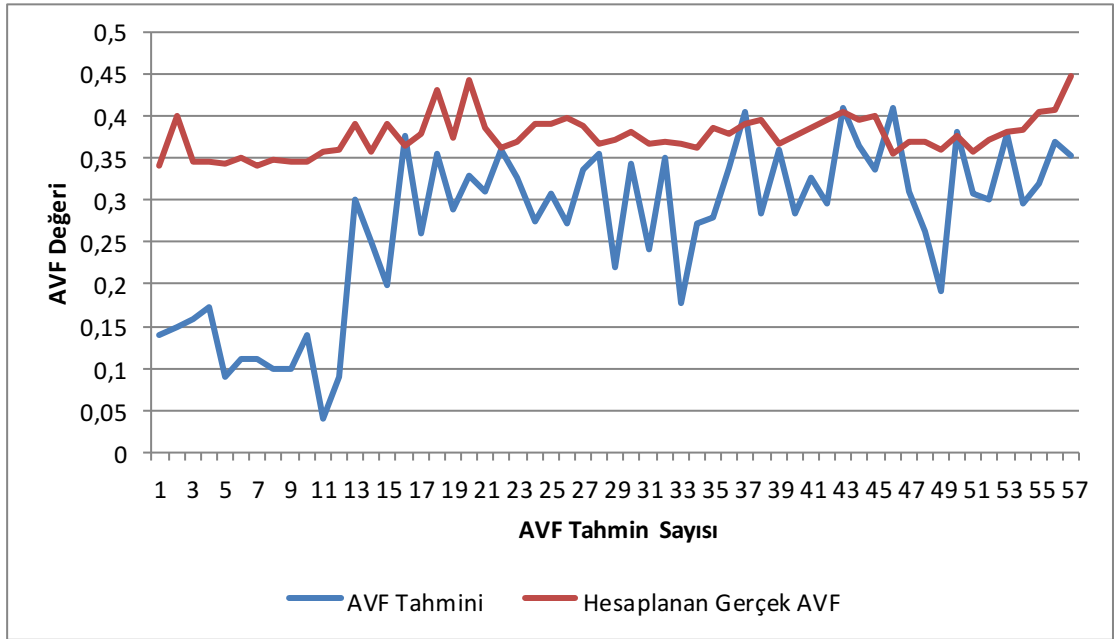


Çizelge 3.3. BZIP2, N: 1000, H: 500, Toplam Buyruk Sayısı: 359581,
AVF Başına Ortalama Buyruk Sayısı: 39953

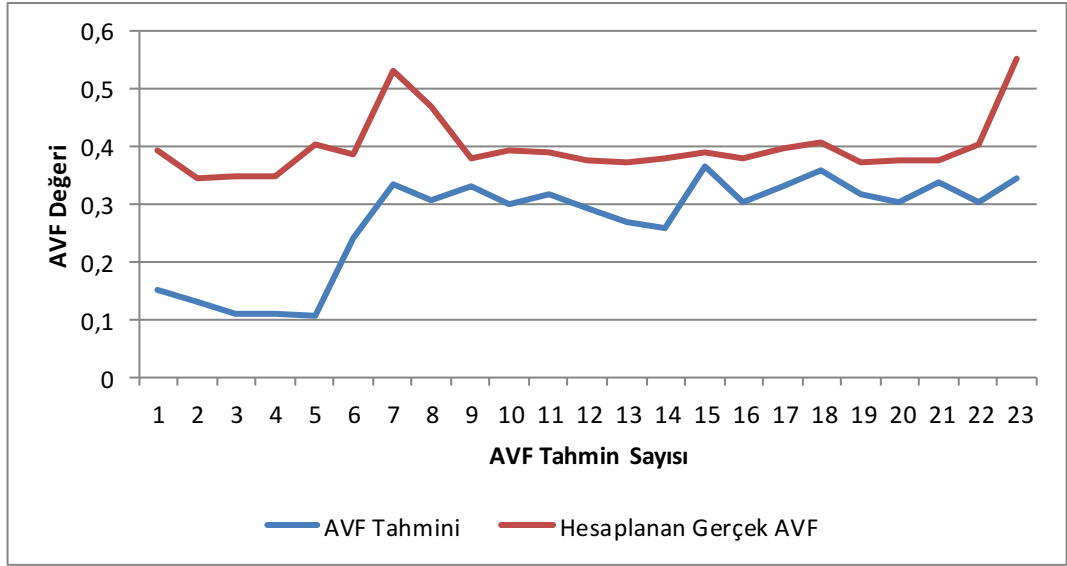


Çizelge 3.4'te GCC buyruk kümesi 100 H ile çalıştırılmıştır. Alınan sonuçta görülen AVF tahminleri BZIP2 buyruk kümesinde 100 H değeri ile alınan sonuçlara benzer şekilde çok belirgin değişimler göstermektedir. Bu çizelgede tahmin edilen AVF değerlerinin gerçek AVF değerleri ile tutarlı olduğunu söylemek biraz zordur. Ancak hem gerçek AVF değerleri hem de tahmin edilen AVF değerleri için genel bir artış olduğunu söylemek mümkündür.

Çizelge 3.4. GCC, N: 1000, H: 100, Toplam Buyruk Sayısı: 5984895,
AVF Başına Ortalama Buyruk Sayısı: 104998



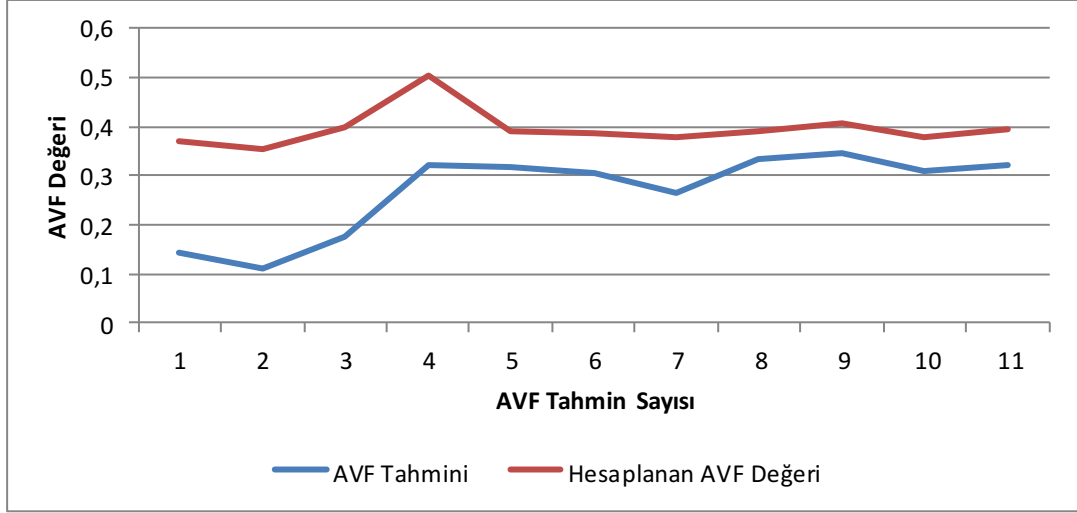
Çizelge 3.5. GCC, N: 1000, H: 250, Toplam Buyruk Sayısı: 6030450,
AVF Başına Ortalama Buyruk Sayısı: 262193



Çizelge 3.5'te GCC buyruk kümesi ile yapılan AVF tahminlerindeki aralıklar daha geniş tutulmuştur. Yani H değeri 250'ye çıkarılmıştır. Bu durumda sonuçların nispeten tutarlı olduğunu söylemek mümkündür. Burada dikkat çekici bir nokta, gerçek AVF değerleri ile yapılan AVF tahminleri arasındaki farkların sürekli azalıyor olmasıdır. Yani gerçek AVF değerleri de tahmin edilen AVF değerleri de sürekli artış gösteriyor olsa da, tahmin edilen AVF değerindeki artış daha fazladır. Bu nedenle sonuçlar arasındaki fark zamanla daralmıştır.

Çizelge 3.6'da GCC buyruk kümesi daha yüksek bir H değeri, yani 500 kullanılarak alınan sonuçlar görülmektedir. Burada iki değer grubu arasındaki tutarlılık daha fazladır. Bazı noktalarda neredeyse birebir artış ve düşüşler görülmektedir. Ancak diğer GCC sonuçlarında olduğu gibi bunda da giderek iki değer grubu arasındaki fark azalmaktadır.

Çizelge 3.6. GCC, N: 1000, H: 500, Toplam Buyruk Sayısı: 5777177,
AVF Başına Ortalama Buyruk Sayısı: 525198



3.4. Değerlendirme

Alınan sonuçlar gösteriyor ki çoklu iş parçacıkları ile çalışma zamanında gerçek AVF değerleri ile tutarlı bir AVF tahmini yapılabilir. Ancak AVF tahminlerinin başarısı verilen parametrelere bağlı olarak değişiyor. Beklenildiği gibi, H değeri arttıkça AVF tahminlerinin gerçek AVF değerleri ile olan tutarlılığı da artıyor. Yani H değeri arttığında sistemin de tutarlılık kazandığı söylenebilir.

Yapılan tahminler çoğunlukla sistemin hataları karşı korunup korunmayacağına karar vermekte yeterlidir. Gerçek AVF değerinin artış gösterdiği, yapılan AVF tahminlerindeki artıştan da anlaşılabilir. Bu şekilde sistem sadece gerektiğinde hatalara karşı korunularak fazla kaynak ve enerji kullanımının önüne geçilmiş olunur.

Ancak çoklu iş parçacıkları ile AVF tahmini yapmanın bazı dezavantajları da vardır. Eğer işlem rastgele değerler üretiyorsa, yani işlem her başladığında zaten farklı çıktılar veriyorsa bu yöntem istenilen sonucu veremeyebilir. Böyle durumlarda AVF tahmini de gerçek AVF değerinden bağımsız olarak %100'e yakın olacaktır.

KAYNAKLAR

- [1] F. Firouzi, A. Azarpeyvand, M. E. Salehi, and S. M. Fakhraie, " Adaptive Fault-Tolerant DVFS with Dynamic Online AVF Prediction", Februray 2012, to be appeared in Elsevier Microelectronics Reliability. (indexed in ISI)
- [2] T.N. Vijaykumar, I. Pomeranz and K. Cheng, "Transient-Fault Recovery Using Simultaneous Multithreading," Proc. 29th Int'l Symp. Computer Architecture (ISCA), 2002.
- [3] S.S. Mukherjee, M. Kontz, and S.K. Reinhardt, "Detailed Design and Evaluation of Redundant Multithreading Alternatives," Proc. 29th Int'l Symp. Computer Architecture, May 2002.
- [4] H. Zhou. A case for fault tolerance and performance enhancement using chip multi-processors. IEEE Computer Architecture Letters, 5(1):22–25, 2006.
- [5] Songjun Pan, Yu Hu, Xiaowei Li: Online Computing and Predicting Architectural Vulnerability Factor of Microprocessor Structures. PRDC 2009: 345-350
- [6] Lide Duan, Lu Peng, Bin Li, "Predicting Architectural Vulnerability on Multithreaded Processors under Resource Contention and Sharing", IEEE Transactions on Dependable and Secure Computing, vol.10, no. 2, pp. 114-127, March-April 2013
- [7] S. Amir Asghari, A. Abdi, H. Taheri, H. Pedram and S. Pourmozaffari, "SEDSR: Soft Error Detection Using Software Redundancy," Journal of Software Engineering and Applications, Vol. 5 No. 9, 2012, pp. 664-670. doi: 10.4236/jsea.2012.59078.
- [8] S.K. Reinhardt and S.S. Mukherjee, "Transient Fault Detection via Simultaneous Multithreading,"Proc. Int'l Symp. Computer Architecture (ISCA), 2000.
- [9] E.D.Lazowska, J.Zahorjan, G.S.Graham, and K.C.Sevcik, "Quantitative System Performance," Prentice-Hall, Inc., E140nglewood Cliffs, New Jersey, 1984.

- [10] Shubhendu S. Mukherjee, Christopher Weaver, Joel Emer, Steven K. Reinhardt, and Todd Austin, "A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor" MICRO 36 Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture, Page 29, 2003.
- [11] S. S. Mukherjee, C. T. Weaver, J. Emer, S. K. Reinhardt, and T. Austin. Measuring architectural vulnerability factors. IEEE MICRO, Dec. 2003.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÇAVUŞ, Mustafa
Uyruğu : T.C.
Doğum tarihi ve yeri : 07.02.1988 Sandıklı
Medeni hali : Bekar
Telefon : 0 (312) 235 46 92
GSM : 0 (506) 260 90 57
e-mail : mcavus@etu.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	TOBB Ekonomi ve Teknoloji Üniversitesi/Bilgisayar	2012
Yüksek Lisans	TOBB Ekonomi ve Teknoloji Üniversitesi/Bilgisayar	2015

Yabancı Dil

İngilizce

Yayımlar

Mustafa Cavus, Hakki Doganer Sumerkan, Osman Seckin Simsek, Hasan Hassan, Abdullah Giray Yaglikci, Oguz Ergin, “GPU Based Parallel Image Processing Library For Embedded Systems” International Conference on Computer Vision And Applications (VISAPP 2014), Lisbon, Portugal, January 2014