

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM2022
Bilgisayar Organizasyonu
Ödev 4

Öğretim Görevlisi
Ali Can KARACA


29 Mayıs 2025

İÇİNDEKİLER

1 İşlemci Bilgileri	1
1.1 İnceleme	2
2 Hesaplamalar	3
2.1 Kod 1 - Boyut: 256	3
2.1.1 Diziler için detaylı inceleme - IJK Formu	4
2.2 Kod 2 - Boyut 256 - IKJ Formu	7
2.3 Kod 3 - Boyut 256 - JIK Formu	9
2.4 64 Boyutlu kodlar için hesaplama	12
2.4.1 Kod 1 - Boyut 64	13
2.4.2 Kod 2 - Boyut 64	14
2.4.3 Kod 3 - Boyut 64	14
3 Süre karşılaştırması	16

İşlemci Bilgileri

AMD Ryzen 7 5800H 8 Core 16 Thread 3.2-4.4 GHz işlemci kullanılmı ştır.

Processor						
Name	AMD Ryzen 7 5800H					
Code Name	Cezanne	Max TDP	45.0 W			
Package	Socket FP6					
Technology	7 nm	Core VID	0.981 V			
Specification	AMD Ryzen 7 5800H with Radeon Graphics					
Family	F	Model	0	Stepping	0	
Ext. Family	19	Ext. Model	50	Revision	CZN-A0	
Instructions	MMX(+), SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A, x86-64, AES, AVX, AVX2, FMA3, SHA					
Caches (Core #0)						
Core Speed		3142.42 MHz		Cache		
Multiplier		x 31.5		L1 Data	8 x 32 KBytes 8-way	
Bus Speed		99.76 MHz		L1 Inst.	8 x 32 KBytes 8-way	
Rated FSB				Level 2	8 x 512 KBytes 8-way	
				Level 3	16 MBytes 16-way	
Selection						
Socket #1		Cores		8	Threads	16

Level 1 cache size ?	8 x 32 KB instruction caches 8 x 32 KB data caches	▼
Level 2 cache size ?	8 x 512 KB	▼
Level 3 cache size	16 MB	▼

MORE DETAILS

The level 1 instruction cache is an 8-way set associative cache with a line size of 64 bytes.
The level 1 data cache is an 8-way set associative cache with a line size of 64 bytes.

MORE DETAILS



The level 2 cache is an 8-way set associative cache with a line size of 64 bytes. The cache contains instructions and data.

MORE DETAILS



The level 3 cache is a 16-way set associative cache with a line size of 64 bytes. The cache is shared between cores. The cache contains instructions and data.

Cache details				
Cache:	L1 data	L1 instruction	L2	L3
Size:	8 x 32 KB	8 x 32 KB	8 x 512 KB	16 MB
Associativity:	8-way set associative	8-way set associative	8-way set associative	16-way set associative
Line size:	64 bytes	64 bytes	64 bytes	64 bytes
Comments:	Direct-mapped	Direct-mapped	Inclusive Direct-mapped	Non-inclusive Direct-mapped Shared between all cores

1.1 İnceleme

İşlemci'nin özelliklerini incelediğimizde ve kodda kullanılan double cinsinden değişkenler için 8 byte kullanıldığını ve cache'nin 64 byte line size'e sahip olduğunu görüyoruz. Bu da veri okunduktan sonraki 8 adet veriyi cache'ye alacağını gösteriyor.

256x256 boyutundaki double dizileri 512 KB yer kapladığı için L1 cache'de bu dizileri tamamen kaydetmenin mümkün olmadığı sonucu ortaya çıkıyor. 64x64 boyutundaki diziler ise 32 KB yer kapladığı için işlemimiz bu verileri ilk kez yazıldığında bir kısmını cache'ye atabileceğini ve daha az miss sayısı ile karışla sağımızı söyleyebiliriz.

2 Hesaplamalar

Teorik olarak yaklaşıklık bir hesaplama yapmak için sadece çok fazla erişim isteyebilecek döngüler hesaplandı. Sabit kodların çok fazla etkisi olmayacağından gerçek sonuçlara yakınlık bekleyebiliriz.

2.1 Kod 1 - Boyut: 256

Dizileri başlatmak için açığımız ilk döngüde:

Dizi[0] - Dizi[65536] aralığında sıralı erişim yapıyoruz. Cache'mizin yapısı gereği ileriki 8 değeri cacheye atacağız.

Bu sebeple:

Dizi[0] MISS

Dizi[1-7] HIT

Dizi[8] MISS

Her 8 değerde bir MISS olacaktır.

HIT RATE = $7/8 = \% 87.5$

196.608 TOTAL WRITE

24.576 MISS WRITE

172.032 HIT WRITE

Ana çarpım döngümüzde: A, B, C okuma ve C yazma erişimleri için her biri için 16.777.216 erişimden:

67.108.864 TOTAL READ

Bu döngüdeki değişkenler için:

$C[\text{dimension} * i + j] += A[\text{dimension} * i + k] * B[\text{dimension} * k + j]$ ifadesinde 9 değişken

için:

$$16.777.216 * 9 = 150.994.944 \text{ READ}$$

En iç döngü için 4 değışkene erişimden:

$$16.777.216 * 4 = 67.108.864 \text{ READ}$$

$$218.103.808 \text{ TOTAL READ}$$

Bu değışkenler cache'de hazır yer alacağından % 100'ye yakın HIT oranı olacaktır.

Ana döngümüzdeki dizilerin erişimide kattığımızda 1.c kodu için toplam yaklaşık olarak:

$$285.279.780 \text{ TOTAL READ}$$

2.1.1 Diziler için detaylı inceleme - IJK Formu

C dizisi için:

$$\text{TOTAL } 16.777.216 \text{ TIMES}$$

$$C[256 * I + J]$$

K döngüsü $C[k] \ C[k] \dots C[k]$ şeklinde sabit okuma yapar MISS etkilenmez.

J döngüsü için $C[0] * 256 \dots C[1] * 256 \dots C[255] * 256 \dots$ şeklinde sıralı okuma yapılır. Bu da her 8 değerinde bir MISS demektir. Toplam 32 MISS yapar.

I döngüsü de katıldığında $C[0] - C[65536]$ arasında sıralı okuma yapılır

Her 1/8'de MISS olacağından 65536 değer için toplam 8192 MISS oluşur.

$$\text{HIT RATE} = \% 99.95$$

$$16.769.024 \text{ READ HITS}$$

$$8.192 \text{ READ MISS}$$

C dizisi önce okunur sonra yazma yapılır bu sebeple aynı adres cache'de yer alacağından WRITE MISS olmaz.

A dizisi için:

$$\text{TOTAL } 16.777.216 \text{ TIMES}$$

$$A[256 * I + K]$$

I döngüsü için MISS sayısı = $256 * J$ döngüsü MISS tekrarı * K döngüsü MISS sayısı

K döngüsü = A[0 + 0-255] = A[0] A[1] A[2] A[255]

K döngüsü HIT RATE = 7/8 = % 87.5

K döngüsü MISS = 32

J döngüsü = A[0-255] A[0-255] A[0-255]

İlk A[0-255] için 32 MISS olacaktır ancak sonrakilerde % 100 HIT olacaktır

Yani J dizisinde sadece bir kez MISS tekrarlanacaktır. Çünkü 0-255 arası elemanlar halihazırda cache'de bulunacaktır.

256 * 1 * 32 = 8192 READ MISS

16.769.024 READ HIT

MISS RATE = % 0.05

HIT RATE = % 99.95

B dizisi için:

TOTAL 16.777.216 TIMES

FOR 256 TIMES

B[256 * K + J]

B[0-65.280 + 0] = B[0] B[256] B[65280]

B[0-65.280 + 1] = B[1] B[257] B[65281]

MISS oranı neredeyse % 100 olacaktır. Dizi erişimleri arasında 256'lık farklar olduğundan dolayı cache sürekli yenilenmek zorunda.

16.777.216 MISS

0 HIT

Ancak işlemcinin L3 boyutu 16 MB olduğundan ve diziler 512 KB olduğundan L3 bu dizileri saklayabilir bu sebeple L3 HIT değeri % 100 olacaktır.

Tüm değerleri topladığımızda elde ettiğimiz teorik ve yaklaşık sonuç:

285.279.780 TOTAL READ

16.793.796 MISS READ

16.973.824 TOTAL WRITE

24.576 MISS WRITE

Gerçek sonuçlarımız ise aşağıdaki gibidir:

```
secs:2.251825
==3567==
==3567== I refs:      783,185,914
==3567== I1 misses:    1,413
==3567== L1i misses:   1,391
==3567== I1 miss rate:  0.00%
==3567== L1i miss rate: 0.00%
==3567==
==3567== D refs:      307,286,171 (288,922,047 rd + 18,364,124 wr)
==3567== D1 misses:   16,894,172 ( 16,869,177 rd +   24,995 wr)
==3567== L1d misses:   25,991 (    1,049 rd +   24,942 wr)
==3567== D1 miss rate:  5.5% (    5.8% +    0.1% )
==3567== L1d miss rate: 0.0% (    0.0% +    0.1% )
==3567==
==3567== LL refs:      16,895,585 ( 16,870,590 rd +   24,995 wr)
==3567== LL misses:    27,382 (    2,440 rd +   24,942 wr)
==3567== LL miss rate:  0.0% (    0.0% +    0.1% )
```

Tablo yapıp karşılaştırsak:

Değer	Teorik Sonuç	Gerçek Sonuç	Aradaki Fark
Total Read	285.279.780	288.922.057	% 1.28
Total Write	16.973.824	18.364.123	% 8.18
L1 Read MISS	16.793.796	16.864.177	% 0.42
L1 Write MISS	24.576	24.995	% 1.7
L1 MISS Ratio	% 5.56	% 5.5	% 1.08

Table 2.1 Teorik ve Gerçek Sonuçların Karşılaştırması

Tablodan anlaşılacağı gibi hesaplanan yaklaşık değer gerçek de ğere çok yakındır. Aradaki farkların sebebi döngüler dışındaki sabit ve etkisi az olan kodların tam olarak hesaplanamamasıdır.

Diğer kodlar için de hesaplama yapılacaktır. Ancak initialize döngüsü ve toplam dizi okuma yazması gibi değerler 256 boyuttaki her kod için aynıdır. Bu sebeple aşağıdaki değerler tekrar yazılmayacaktır:

285.279.780 TOTAL READ

16.973.824 TOTAL WRITE

24.576 MISS WRITE

2.2 Kod 2 - Boyut 256 - IKJ Formu

C dizisi için:

TOTAL 16.777.216 TIMES

C[256 * I + J]

I döngüsü MISS sayısı = '256 * K döngüsü MISS tekrarı * J döngüsü MISS sayısı'

J döngüsü = C[0 + 0-255] = C[0] C[1] C[2] ... C[255] ,sekinde sıralı devam eder

J döngüsü HIT oranı = 7/8 = % 87.5

J döngüsü MISS sayısı = 32

K döngüsü ise: C[0] C[0] ... C[0] ,sekinde 256 kez tekrarlarBu da MISS olu ,sturmaz.

Dolayısıyla MISS sayısı = 256 * 32 * 1 = **8.192**

HIT Oranı = 7/8 = % 99.95

16.769.024 READ HITS

8.192 READ MISS

C dizisi önce okunur sonra yazma yapılır bu sebeple aynı adres cache'de yer alacağından WRITE MISS olmaz.

A dizisi için:

TOTAL 16.777.216 TIMES

A[256 * I + K]

J döngüsü için 256 kez:

A[0 + 0] = A[0] A[0] A[0] A[0] ,sekinde sabit okuma yapılacaktır.

D1 ,staki K döngüsü ise dizi indeksini sırasıyla arttırır

K döngüsü için a ,şğıdaki ,sekinde A[255]'e gidecek ,sekinde okuma yapılır.

256 * A[0] ... 256 * A[1] ... 256 * A[7] ... 256 * A[255]

Bu sebeple K döngüsü içinde toplamda 32 MISS olu ,sur.

I döngüsü ise 256 kez bu MISS'leri tekrarlar ve 256 * 32 = **8192** MISS olu ,sur.

A dizisi için sonuç:

8192 READ MISS

16.769.024 READ HIT

MISS RATE = % 0.050

HIT RATE = % 99.95

B dizisi için:

TOTAL 16.777.216 TIMES

FOR 256 TIMES

$B[256 * K + J]$

J döngüsü a, şğıdaki gibi sıralı devam ettirecektir:

$B[0 + 0-255] = B[0] B[1] B[2] \dots B[255]$

Sıralı ilerlediğı için HIT oranı 7/ 8 olur. Toplamda 32 MISS gerçekte, sir.

K döngüsü ise: $B[256 + 0-255] \dots B[512 + 0-255] \dots = B[256] B[257] \dots B[511]$
 $B[512] \dots$

Şeklinde aynı sıralı ilerlemeyi sürdürür. Bu sebeple oran yine 7/ 8 olacaktır.

K döngüsü içinde toplamda 8192 MISS gerçekte, sir.

I döngüsü ise a, şğıdaki, şekilde bu döngüleri 256 kez etki ettirmeden devam ettirir:

$B[0-65536] B[0-65536] \dots = 256 \text{ TIMES}$

256 kez tekrarladığı için MISS sayısı = $256 * 8192 = 2.097.152$

B dizisi için sonuç:

2.097.152 READ MISS

14.680.064 READ HIT

HIT RATE = 7/ 8 = % 87.5

Tüm değerleri topladığımızda elde ettiğimiz teorik ve yakla, sık sonuç:

285.279.780 TOTAL READ

2.113.536 MISS READ

16.973.824 TOTAL WRITE

24.576 MISS WRITE

Gerçek sonuçlarımız ise aşağıdaki gibidir:

```
secs:2.255238
==3710==
==3710== I refs:          783,185,925
==3710== I1 misses:       1,415
==3710== L1i misses:      1,393
==3710== I1 miss rate:    0.00%
==3710== L1i miss rate:   0.00%
==3710==
==3710== D refs:          307,286,174 (288,922,049 rd + 18,364,125 wr)
==3710== D1 misses:       2,140,124 ( 2,115,129 rd +    24,995 wr)
==3710== L1d misses:      25,991 (    1,049 rd +    24,942 wr)
==3710== D1 miss rate:    0.7% (    0.7% +    0.1% )
==3710== L1d miss rate:   0.0% (    0.0% +    0.1% )
==3710==
==3710== LL refs:          2,141,539 ( 2,116,544 rd +    24,995 wr)
==3710== LL misses:        27,384 (    2,442 rd +    24,942 wr)
==3710== LL miss rate:    0.0% (    0.0% +    0.1% )
```

Tablo yapıp karşılaştırsak:

Değer	Teorik Sonuç	Gerçek Sonuç	Aradaki Fark
Total Read	285.279.780	288.922.049	% 1.28
Total Write	16.973.824	18.364.125	% 8.18
L1 Read MISS	2.113.536	2.115.129	% 0.075
L1 Write MISS	24.576	24.995	% 1.7
L1 MISS Ratio	% 0.71	% 0.7	% 1.41

Table 2.2 Teorik ve Gerçek Sonuçların Karşılaştırması

MISS sayıları ve oranları neredeyse birebir çıkmıştır. Bu da hesaplamaların doğru olduğunu göstermektedir.

2.3 Kod 3 - Boyut 256 - JIK Formu

C dizisi için:

K döngüsünde aşağıdaki gibi sabit erişim işlemi yapılmıyorsa MISS etkilenmez.:

$$C[0 + 0] = C[0] \quad C[0] \quad C[0] \quad \dots * 256$$

I döngüsünde ise aşağıdaki gibi 256 artarak devam edilir:

$$C[0-65280 + 0] = C[0] \quad \dots C[256] \quad \dots C[512]$$

Bu sebeple MISS oranı % 100 olur. Toplamda 256 MISS olur.

J döngüsü ise $[0 - 65280] \rightarrow [1 - 65281] \dots$,sekinde döngüleri ayrık devam ettirir.
MISS oranı yine % 100'dür.

$$C[0-65280 + 0-255] = C[0] C[256] \dots C[65280]$$

Devam eder...

$$C[1] C[257] C[513] \dots$$

$$\text{Toplamda } 256 * 256 * 1 = 65536 \text{ MISS olur, sur.}$$

C dizisi için sonuç:

$$\text{MISS RATE} = \% 0.39$$

$$\text{HIT RATE} = \% 99.61$$

$$\text{TOTAL MISS COUNT} = 65536$$

$$\text{TOTAL HIT COUNT} = 16.711.680$$

C dizisi önce okunur sonra yazma yapılır bu sebeple aynı adres cache'de yer alacağından WRITE MISS olmaz.

A dizisi için:

K döngüsü a,şğıdaki gibi sıralı devam eder:

$$C[0 + 0-255] = C[0] C[1] C[2] \dots C[255]$$

$$\text{K döngüsü HIT oranı} = 7/8$$

$$\text{K döngüsü MISS sayısı} = 32$$

I döngüsü ise K döngüsünü a,şğıdaki gibi devam ettirir.

$$A[0-65536] = A[0] \dots A[65536]$$

Bu sebeple HIT oranı değışmez.

$$\text{MISS sayısı I döngüsü için, } 65536 * 7/8 = 8192 \text{ 'dir.}$$

J döngüsü ise dizi indeksini değıştirmez. Sadece iç döngülerin 256 kez devam etmesine dolayısıyla 256 kez MISS olan yerleri tekrarlar

$$A[0-65536] A[0-65536] \dots * 256$$

Bu sebeple MISS oranı % 100'dür.

$$\text{Toplam MISS sayısı} = 256 * 256 * 32 = 2.097.152$$

A dizisi için sonuç:

MISS RATE = % 12.5

HIT RATE = % 87.5

MISS COUNT = 2.097.152

HIT COUNT = 14.680.064

B dizisi için:

K döngüsünde a, şğıdaki gibi ayrık ilerlenir:

$B[0-65280 + 0] = B[0] \ B[256] \ B[512] \ ... \ B[65280]$

Bu sebeple:

K döngüsü MISS oranı = % 100

K döngüsü MISS sayısı = 256

I döngüsü ise indeksi etkilemez ancak bu döngüleri 256 kez tekrarladğı için MISS'ler de 256 kez tekrarlanır.

$B[0-65280] \ ... \ B[0-65280] \ ... \ * \ 256$

Bu sebeple:

I döngüsü MISS oranı = % 100

I döngüsü MISS sayısı = $256 * 256 = 65536$

J döngüsü ise değeri ayrı ayrı olarak bir arttırarak ilerletir. Yine çok aralıklı ilerlendiğı için cache'de verimli saklanamaz. MISS oranı % 100 olur.

$B[0-65280 + 0-256] = B[0-65280] \ ... \ B[1-65281] \ ... \ B[2-65282] \ ... \ B[65536]$

B dizisi için sonuç:

MISS oranı = % 100

MISS sayısı = $256 * 256 * 256 = 16.777.216$

Tüm değeri topladığımızda elde ettiğimiz teorik ve yaklaşık sonuç:

285.279.780 TOTAL READ

18.939.904 MISS READ

16.973.824 TOTAL WRITE

24.576 MISS WRITE

Gerçek sonuç ise aşağıdaki gibidir:

```
secs:2.303819
==3728==
==3728== I refs:      783,185,928
==3728== I1 misses:    1,416
==3728== L1i misses:   1,394
==3728== I1 miss rate:  0.00%
==3728== L1i miss rate: 0.00%
==3728==
==3728== D refs:      307,286,175 (288,922,050 rd + 18,364,125 wr)
==3728== D1 misses:   18,966,490 ( 18,941,495 rd +   24,995 wr)
==3728== L1d misses:   25,991 (   1,049 rd +   24,942 wr)
==3728== D1 miss rate:  6.2% (   6.6% +   0.1% )
==3728== L1d miss rate: 0.0% (   0.0% +   0.1% )
==3728==
==3728== LL refs:      18,967,906 ( 18,942,911 rd +   24,995 wr)
==3728== LL misses:    27,385 (   2,443 rd +   24,942 wr)
==3728== LL miss rate:  0.0% (   0.0% +   0.1% )
```

Tablo yapıp karşılaştırsak:

Değer	Teorik Sonuç	Gerçek Sonuç	Aradaki Fark
Total Read	285.279.780	288.922.050	% 1.28
Total Write	16.973.824	18.364.125	% 8.18
L1 Read MISS	18.939.904	18.941.495	% 0.0084
L1 Write MISS	24.576	24.995	% 1.7
L1 MISS Ratio	% 6.27	% 6.17	% 1.59

Table 2.3 Teorik ve Gerçek Sonuçların Karşılaştırması

Sonuçlar daha da yakın çıkmıştır. READ MISS değerinde % 0.0084 gibi önemsiz bir fark bulunmuştur. Bu da ölçümlerin tutarlılığını kanıtlar niteliktedir.

2.4 64 Boyutlu kodlar için hesaplama

Aynı amaçlar tüm kodlar için hesaplanmıştır fakat sadece sonuçlar gösterilecektir. Bu kısımda işlemcinin L1 cache'si 32 KB boyutunda ve her bir dizimiz de 32 KB boyutundadır. Bu sebeple her dizinin neredeyse üçte biri cache'de yenilenmeden durabilmektedir. Bunun sonucunda tüm gerçek çıktıların teorik ölçümlerden daha iyi sonuç vermiştir.

2.4.1 Kod 1 - Boyut 64

Kod 1, boyut 64 için hesaplanan sonuçlar aşağıdaki gibidir:

4.457.496 TOTAL READ

262.403 MISS READ

265.216 TOTAL WRITE

1.536 MISS WRITE

Gerçek sonuç ise aşağıdaki gibidir:

```
secs:0.040568
==3738==
==3738== I refs:          12,925,590
==3738== I1 misses:      1,440
==3738== L1i misses:     1,417
==3738== I1 miss rate:    0.01%
==3738== L1i miss rate:   0.01%
==3738==
==3738== D refs:          5,098,211 (4,724,074 rd + 374,137 wr)
==3738== D1 misses:      52,131 ( 50,175 rd + 1,956 wr)
==3738== L1d misses:     2,956 ( 1,055 rd + 1,901 wr)
==3738== D1 miss rate:    1.0% ( 1.1% + 0.5% )
==3738== L1d miss rate:   0.1% ( 0.0% + 0.5% )
==3738==
==3738== LL refs:         53,571 ( 51,615 rd + 1,956 wr)
==3738== LL misses:       4,373 ( 2,472 rd + 1,901 wr)
==3738== LL miss rate:    0.0% ( 0.0% + 0.5% )
```

Tablo yapıp karşılaştırsak:

Değer	Teorik Sonuç	Gerçek Sonuç	Aradaki Fark
Total Read	4.457.496	4.724.074	% 5.98
Total Write	265.216	374.137	% 41.02
L1 Read MISS	262.403	50.175	% 80.88
L1 Write MISS	1.536	1.956	% 27.34
L1 MISS Ratio	% 5.59	% 1.0	% 82.1

Table 2.4 Teorik ve Gerçek Sonuçların Karşılaştırması

Görüldüğü gibi sonuçlar oldukça farklı çıkmıştır. Bunun sebebi dizilerin boyutunun küçülmesi ile cache’de yenileme yapılmadan daha fazla veri tutulabilmesidir.

Bazı değerlerde fark görülmesinin sebebi ise döngülerden kaynaklanmayan sabit değerlerin boyut azaldıkça daha fazla etkiye sahip olmasıdır.

Bu da bize 64 boyutun teorik hesaplama için yetersiz olduğu çıkarımını vermektedir.

2.4.2 Kod 2 - Boyut 64

Kod 2, boyut 64 için hesaplanan sonuçlar aşağıdaki gibidir:

4.457.496 TOTAL READ

33.792 MISS READ

265.216 TOTAL WRITE

1.536 MISS WRITE

Gerçek sonuç ise aşağıdaki gibidir:

```
secs:0.038593
==3747==
==3747== I refs:          12,925,599
==3747== I1 misses:       1,440
==3747== L1i misses:      1,417
==3747== I1 miss rate:    0.01%
==3747== L1i miss rate:   0.01%
==3747==
==3747== D refs:          5,098,214 (4,724,077 rd + 374,137 wr)
==3747== D1 misses:       13,692 ( 11,736 rd + 1,956 wr)
==3747== L1d misses:      2,956 ( 1,055 rd + 1,901 wr)
==3747== D1 miss rate:    0.3% ( 0.2% + 0.5% )
==3747== L1d miss rate:   0.1% ( 0.0% + 0.5% )
==3747==
==3747== LL refs:         15,132 ( 13,176 rd + 1,956 wr)
==3747== LL misses:       4,373 ( 2,472 rd + 1,901 wr)
==3747== LL miss rate:    0.0% ( 0.0% + 0.5% )
```

Tablo yapıp karşılaştırsak:

Değer	Teorik Sonuç	Gerçek Sonuç	Aradaki Fark
Total Read	4.457.496	4.724.077	% 5.98
Total Write	265.216	374.137	% 41.02
L1 Read MISS	33.792	11.736	% 65.3
L1 Write MISS	1.536	1.956	% 27.34
L1 MISS Ratio	% 0.75	% 0.3	% 60

Table 2.5 Teorik ve Gerçek Sonuçların Karşılaştırması

Benzer sonuçlar gözlemlenmiştir.

2.4.3 Kod 3 - Boyut 64

Kod 3, boyut 64 için hesaplanan sonuçlar aşağıdaki gibidir:

4.457.496 TOTAL READ

295.936 MISS READ

265.216 TOTAL WRITE

1.536 MISS WRITE

Gerçek sonuç ise aşağıdaki gibidir:

```
secs:0.041667
==3756==
==3756== I refs:      12,925,610
==3756== I1 misses:    1,442
==3756== L1i misses:   1,419
==3756== I1 miss rate:  0.01%
==3756== L1i miss rate: 0.01%
==3756==
==3756== D refs:      5,098,217 (4,724,079 rd + 374,138 wr)
==3756== D1 misses:    76,292 ( 74,336 rd + 1,956 wr)
==3756== L1d misses:    2,956 ( 1,055 rd + 1,901 wr)
==3756== D1 miss rate:  1.5% ( 1.6% + 0.5% )
==3756== L1d miss rate: 0.1% ( 0.0% + 0.5% )
==3756==
==3756== LL refs:      77,734 ( 75,778 rd + 1,956 wr)
==3756== LL misses:    4,375 ( 2,474 rd + 1,901 wr)
==3756== LL miss rate:  0.0% ( 0.0% + 0.5% )
```

Tablo yapıp karşılaştırsak:

Değer	Teorik Sonuç	Gerçek Sonuç	Aradaki Fark
Total Read	4.457.496	4.724.077	% 5.98
Total Write	265.216	374.137	% 41.02
L1 Read MISS	295.936	74.336	% 74.9
L1 Write MISS	1.536	1.956	% 27.34
L1 MISS Ratio	% 6.3	% 1.5	% 76.19

Table 2.6 Teorik ve Gerçek Sonuçların Karşılaştırması

Tüm 64 boyutluk sonuçlar cache boyutunun önemini ortaya koymaktadır.

Boyutlar IR sayısını değiştirse de MISS sayıları değişmemiştir. Bu da bu Instruction MISS'lerinin sabit kodlardan kaynaklandığı tezimizi doğrulamaktadır.

3

Süre kar ,sıla ,stırması

Süre olarak çıktılar a ,şındaki gibidir (VM ortamında i ,slemci yava ,s çağırıldan ve fark gözlemlenmeyecek kadar az olduğundan gerçek ortamda denenmi ,stir):

Kod 1 - Boyut 256: **0.070983 saniye**

Kod 2 - Boyut 256: **0.051713 saniye**

Kod 3 - Boyut 256: **0.072605 saniye**

Kod 1 - Boyut 64: **0.001016 saniye**

Kod 2 - Boyut 64: **0.000906 saniye**

Kod 3 - Boyut 64: **0.001028 saniye**

Sonuç olarak MISS oranlarını ve sayılarını kar ,sıla ,stırırsak:

Kod 3 > Kod 1 > Kod 2

Cache MISS oranları arttıkça programın süresi de artmı ,stır. Bu da cache için daha optimize i ,slemlerin performans farkını ortaya koymaktadır.

Örneğin ikinci kod ilk koda göre % **27.14** daha hızlıdır.

L1 Instruction Cache için analiz edelim (İşlemci 32 KB instruction cache'ye sahip):

256 Boyutu İçin

Küçük atamaları atlarsak ve döngüler üzerinden ilerleyerek yaklaşık olarak hesaplırsak:

- **Initialize döngüsü:**
 - I döngüsü: 1.792
 - J döngüsü: 458.752
 - Döngü içi işlemler: 2.621.440
- **Ana döngü:**
 - 1. döngü: 1.792
 - 2. döngü: 458.752
 - 3. döngü: 117.440.512
- **Döngü içi toplam işlemler:** 452.984.832
- **Toplam işlem sayısı:** 573.967.872

Kod oldukça kısa olduğundan 32 KB'lık L1 instruction cache'ye yüklenebilir. Bu yüzden her kod ve her boyut için MISS sabit çıkmıştır. 256 boyut için 1416 olarak gözlemlendi.

Gerçek çıktı sonucunda yaklaşık 783.185.914 işlem çıktısı gözlemlenmiştir. Daha derin seviyeye inip detaylı analiz yapamadığımız için ve random fonksiyonu gibi fonksiyonlar da kullanıldığı için tam olarak gerçek sonuca yaklaşılamadık.

64 Boyutu İçin

Yaklaşık işlem sayıları:

- **Initialize döngüsü:**
 - I döngüsü: 448
 - J döngüsü: 28.672
 - Döngü içi işlemler: 163.840
- **Ana döngü:**
 - 1. döngü: 448
 - 2. döngü: 28.672
 - 3. döngü: 1.835.008
- **Döngü içi toplam işlemler:** 6.815.744
- **Toplam işlem sayısı:** 8.708.992

Gerçek çıktı sonucu 12.925.590 olarak gözlemlenmiştir. Bir nebze de olsa yaklaşılabildik fakat aynı şekilde tam olarak analiz edemedik.

Instruction MISS değeri ise 1442 olarak gözlemlendi. Tekrardan boyutun Instruction MISS'lerine etkisi olmadığı kanıtlandı.