



**دانشگاه صنعتی امیرکبیر**  
(پلی تکنیک تهران)

گزارش پروژه نهایی شبیه سازی معماری کامپیوتر پیشرفته

## 1. تمرین اول: Parsec

### 1.1 Build GEM5

ابتدا GEM5 ای که فایل آن در پوشه تمرین قرار داشت، را با ISA ALPHA روی اوپونتوی 16.04 بیلد کردیم.

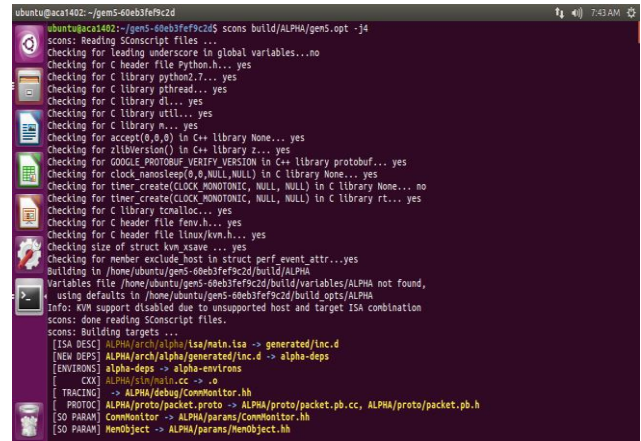


Figure 1 بیلد gem5 با ISA آلفا

### 1.2 Parsec Configuration

سپس برای استفاده از بنچ مارک Parsec، مراحل زیر را به ترتیب انجام دادیم<sup>1</sup>:

- ابتدا با دستور زیر یک پوشه به نام `full_system_images` برای ذخیره `image` بنچ مارک Parsec ایجاد می‌کنیم:

`mkdir full_system_image`

```
ubuntu@aca1402:~/gem5-60eb3fef9c2d$ mkdir full_system_images
ubuntu@aca1402:~/gem5-60eb3fef9c2d$ cd full_system_images
```

- در گام بعدی فایل‌های اولیه سیستم را دانلود کرده و آن را از حالت فشرده خارج می‌کنیم:

```
ubuntu@aca1402:~/gem5-60eb3fef9c2d/full_system_images$ tar jxvf m5_system_2.0b3.tar.bz2
m5_system_2.0b3/
m5_system_2.0b3/binaries/
m5_system_2.0b3/binaries/vmlinux
m5_system_2.0b3/binaries/ts_osfpal
m5_system_2.0b3/binaries/console
m5_system_2.0b3/disks/
m5_system_2.0b3/disks/linux-latest.img
m5_system_2.0b3/disks/linux-bigswap2.img
ubuntu@aca1402:~/gem5-60eb3fef9c2d/full_system_images$ mv m5_system_2.0b3 system
```

برای این مرحله از کدهای زیر استفاده شده است :

<sup>1</sup> برای این بخش از راهنمایی سایت زیر استفاده شده است:

[Configure and Run PARSEC-2.1 Benchmark in Gem5](https://pfzuo.github.io/Configure-and-Run-PARSEC-2.1-Benchmark-in-Gem5/)  
([pfzuo.github.io](https://pfzuo.github.io/))

دانلود فایل های اولیه سیستم :

wget

[http://www.m5sim.org/dist/current/m5\\_system\\_2.0b3.tar.bz2](http://www.m5sim.org/dist/current/m5_system_2.0b3.tar.bz2)

باتوجه به خرابی لینک بالا، از فایلی که توسط تدریس‌یاران محترم در کانال درسی قرار داده شده، استفاده شد.

خارج کردن فایل از حالت فشرده:

`tar jxvf m5_system_2.0b3.tar.bz2`

- تا این مرحله، یک پوشه با محتویات فایل های زیر در سیستم ایجاد شد:
- در پوشه `system/binaries` فایل های:
- `Consol` : این فایل یک شبیه‌ساز ترمینال است که برای تعامل با سیستم عامل شبیه‌سازی شده استفاده می‌شود.

`Ts_osfpal` : این فایل یک برنامه تست است که برای بررسی عملکرد سیستم عامل شبیه‌سازی شده در شرایط مختلف استفاده می‌شود.

`Vmlinux` : این فایل هسته سیستم عامل `Linux` است که شامل کد اصلی سیستم عامل بوده و برای بوت شدن سیستم عامل شبیه‌سازی شده استفاده می‌شود.

- در پوشه `system/disk` فایل های:

`Linux-bigswap2.img`: این فایل یک تصویر دیسک مجازی است که شامل سیستم عامل `Linux` و فضای `swap` اضافی است.

`Linux-latest.img`: این فایل یک تصویر دیسک مجازی است که شامل سیستم عامل `Linux` بدون فضای `swap` اضافی است.

- سپس در گام بعدی، بنچ مارک `parsec` را دانلود کرده و فایل `vmlinux` و `ts_osfpal` را به جای فایل های پوشه `system` قرار می‌دهیم.

از دستورات زیر در این مرحله استفاده شده است:

- دانلود vmlinux جدید از لینک زیر:

[http://www.cs.utexas.edu/~parsec\\_m5/vmlinux\\_2.6.27-gcc\\_4.3.4](http://www.cs.utexas.edu/~parsec_m5/vmlinux_2.6.27-gcc_4.3.4)

- قرار دادن این فایل به جای vmlinux با دستور زیر:

```
rm vmlinux
```

- تغییر نام فایل جدید به vmlinux با دستور زیر:

```
mv vmlinux_2.6.27-gcc_4.3.4 vmlinux
```

- دانلود فایل tsb\_osfpal از لینک زیر:

[http://www.cs.utexas.edu/~parsec\\_m5/tsb\\_osfpal](http://www.cs.utexas.edu/~parsec_m5/tsb_osfpal)

- قرار دادن این فایل به جای ts\_osfpal با دستور زیر:

```
rm ts_osfpal
```

- سپس با دستور زیر، فایل pal جدید را به ts\_osfpal

تغییر نام می‌دهیم:

```
mv tsb_osfpal ts_osfpal
```

- دانلود image parsec2-1 از لینک زیر (در

دایرکتوری پوشه disks):

[http://www.cs.utexas.edu/~parsec\\_m5/linux-parsec-2-1-m5-with-test-inputs.img.bz2](http://www.cs.utexas.edu/~parsec_m5/linux-parsec-2-1-m5-with-test-inputs.img.bz2)

- سپس با دستور زیر فایل دانلود شده را decompress

می‌کنیم:

```
bzip2 -d linux-parsec-2-1-m5-with-test-inputs.img.bz2
```

```
~/gen5-60eb3fef9c2d/full_system_images/system/binaries$ rm vmlinux
~/gen5-60eb3fef9c2d/full_system_images/system/binaries$ mv vmlinux_2.6.27-gcc_4.3.4 vmlinux
~/gen5-60eb3fef9c2d/full_system_images/system/binaries$ rm ts_osfpal
~/gen5-60eb3fef9c2d/full_system_images/system/binaries$ mv tsb_osfpal ts_osfpal
~/gen5-60eb3fef9c2d/full_system_images/system/binaries$ cd ../disks/
~/gen5-60eb3fef9c2d/full_system_images/system/disk$ bzip2 -d linux-parsec-2-1-m5-with-test-inputs.img.bz2
```

- در گام بعدی از تکست ادیتور nano برای ویرایش

path در syspaths.py و benchmarks.py

استفاده می‌کنیم.

- برای syspaths.py به صورت زیر:

```
nano ./configs/common/SysPaths.py
```

قبل از ویرایش:

```
path =
['/dist/m5/system', '/n/poolfs/z/dist/m5/system']
```

بعد از ویرایش:

```
path= ['/dist/m5/system', '/home/ubuntu/gen5-60eb3fef9c2d/full_system_images/system']
```

```
path = [ '/dist/m5/system', '/home/ubuntu/gen5-60eb3fef9c2d/full_system_images/system']
```

- برای benchmarks.py به صورت زیر:

```
nano ./configs/common/benchmarks.py
```

قبل از ویرایش:

```
elif buildEnv['TARGET_ISA'] == 'alpha':
return env.get('LINUX_IMAGE', disk('linux-
latest.img'))
```

بعد از ویرایش:

```
elif buildEnv['TARGET_ISA'] == 'alpha':
return env.get('LINUX_IMAGE', disk('linux-
parsec-2-1-m5-with-test-inputs.img'))
```

```
elif buildEnv['TARGET_ISA'] == 'alpha':
return env.get('LINUX_IMAGE', disk('linux-parsec-2-1-m5-with-test-inputs.img'))
```

- در ادامه ابزار parsec که وظیفه ی ساخت اسکریپت

برای تست را دارد، از لینک زیر دانلود می‌کنیم:

[http://www.cs.utexas.edu/~parsec\\_m5/TR-09-32-parsec-2.1-alpha-files.tar.gz](http://www.cs.utexas.edu/~parsec_m5/TR-09-32-parsec-2.1-alpha-files.tar.gz)

```
ubuntu@aca1402:~/gem5-60eb3fef9c2d$ build/ALPHA/gem5.opt ./configs/example/fs.py -n 1 --script=./TR-09-32-parsec-2.1-alpha-files/blackscholes_1c_test.rcs --cpu-type=AtomicSimpleCPU --caches --l1i_size=16kB --l1d_size=16kB --l1i_assoc=4 --l1d_assoc=4 --l2cache --l2_size=256kB --l2_assoc=4 -F 5000000000
gem5 Simulator System.  http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 compiled Feb  6 2024 04:32:45
gem5 started Feb  6 2024 04:59:03
gem5 executing on aca1402
command line: build/ALPHA/gem5.opt ./configs/example/fs.py -n 1 --script=./TR-09-32-parsec-2.1-alpha-files/blackscholes_1c_test.rcs --cpu-type=AtomicSimpleCPU --caches --l1i_size=16kB --l1d_size=16kB --l1i_assoc=4 --l1d_assoc=4 --l2cache --l2_size=256kB --l2_assoc=4 -F 5000000000

/home/ubuntu/gem5-60eb3fef9c2d/configs/common/CacheConfig.py:48: SyntaxWarning: import * only allowed at module level
def config_cache(options, system):
Global frequency set at 1000000000000 ticks per second
warn: DRAM device capacity (512 Mbytes) does not match the address range assigned (512 Mbytes)
info: kernel located at: /home/ubuntu/gem5-60eb3fef9c2d/full_system_images/system/binaries/wlntux
listening for system connection on port 3456
0: system.tsunami.io.rtc: Real-time clock set to Thu Jan  1 00:00:00 2009
0: system.remote_gdb.listener: listening for remote gdb #0 on port 7000
Switch at instruction count:5000000000
Info: Entering event queue @ 0. Starting simulation...
warn: Prefetch instructions to Alpha do not do anything
```

فایل خروجی با اسم statsblackscholes در پوشه Q1 تمرین قرار داده شده است.  
باتوجه به اینکه عملیات شبیه سازی چهار بار انجام شده، مقادیر پارامترهای خواسته شده در هر چهار راند عبارتند از :

**blackscholes 1 Table**

Round	Demand parameter	Simulation value
1	sim_insts	97989385
	Committed_Insts	729912
	Issued_Insts	?
	icache.overall_miss_rate	0.011686
	dcache.overall_miss_rate	0.082124
	l2.overall_miss_rate	0.592118
	cpu.dcache.read/cpu.dcache.write	133493/ 124835= 1.06
	Mem.reads/mem.writes	17834/ 8747= 2.03
2	sim_insts	10003617 0
	Committed_Insts	1305811
	Issued_Insts	?
	icache.overall_miss_rate	0.023952
	dcache.overall_miss_rate	0.054788
	l2.overall_miss_rate	0.337144
	cpu.dcache.read/cpu.dcache.write	248420/ 161852= 1.5
	Mem.reads/mem.writes	18332/ 19936= 0.91

سپس با دستور زیر، فایل دانلود شده را از حالت فشرده خارج می‌کنیم:

`tar zxvf TR-09-32-parsec-2.1-alpha-files.tar.gz`

```
ubuntu@aca1402:~/gem5-60eb3fef9c2d$ tar zxvf TR-09-32-parsec-2.1-alpha-files.tar.gz
TR-09-32-parsec-2.1-alpha-files/
TR-09-32-parsec-2.1-alpha-files/gcc-4.2.4-glibc-2.3.6-tls.dat
TR-09-32-parsec-2.1-alpha-files/version-info.h
TR-09-32-parsec-2.1-alpha-files/parsec-2.1-extra-alpha-packages.tar.gz
TR-09-32-parsec-2.1-alpha-files/unpack-parsec-2.1-extra-alpha-packages
TR-09-32-parsec-2.1-alpha-files/parsec-2.1-alpha-diffs.patch
TR-09-32-parsec-2.1-alpha-files/hooks/
```

تا اینجا مراحل لازم برای نصب و پیکربندی Parsec انجام شد. در گام های بعدی برای سه نمونه خواسته شده از برنامه های بسته محک parsec مراحل زیر انجام شد.

### Blackscholes 1.3

ابتدا در دایرکتوری پوشه TR-09-32-parsec با دستور زیر برنامه مورد نظر (در اینجا Blackscholes) تعریف و انتخاب می‌شود:

`./writescrpts.pl blackscholes 1`

```
ubuntu@aca1402:~/gem5-60eb3fef9c2d$ cd TR-09-32-parsec-2.1-alpha-files/
ubuntu@aca1402:~/gem5-60eb3fef9c2d/TR-09-32-parsec-2.1-alpha-files$ ./writescrpts.pl blackscholes 1
```

سپس gem5 را با دستور زیر برای این برنامه به صورت تک هسته اجرا می‌کنیم:

```
build/ALPHA/gem5.opt
./configs/example/fs.py -n 1
--script=./TR-09-32-parsec-2.1-alpha-files/blackscholes_1c_test.rcs
--cpu-type=AtomicSimpleCPU
--caches
--l1i_size=16kB
--l1d_size=16kB
--l1i_assoc=4
--l1d_assoc=4
--l2cache
--l2_size=256kB
--l2_assoc=4
-F 5000000000
```

```
--l1i_assoc=4
--l1d_assoc=4
--l2cache
--l2_size=256kB
--l2_assoc=4
-F 5000000000
```

```
ubuntu@aca1402:~/gem5-60eb3fef9c2d/TR-09-32-parsec-2.1-alpha-files$ cd ..
ubuntu@aca1402:~/gem5-60eb3fef9c2d$ build/ALPHA/gem5.opt ./configs/example/fs.py -n 1 --script=./TR-09-32-parsec-2.1-alpha-files/bodytrack_1c_test.rcs --cpu-type=AtomicSimpleCPU --caches --l1i_size=16kB --l1d_size=16kB --l1i_assoc=4 --l1d_assoc=4 --l2cache --l2_size=256kB --l2_assoc=4 -F 5000000000
gem5 simulator system: http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 compiled Feb  6 2024 04:32:45
gem5 started Feb  9 2024 08:12:18
gem5 executing on aca1402
Command line: build/ALPHA/gem5.opt ./configs/example/fs.py -n 1 --script=./TR-09-32-parsec-2.1-alpha-files/bodytrack_1c_test.rcs --cpu-type=AtomicSimpleCPU --caches --l1i_size=16kB --l1d_size=16kB --l1i_assoc=4 --l1d_assoc=4 --l2cache --l2_size=256kB --l2_assoc=4 -F 5000000000
Global frequency set at 100000000000 ticks per second
warn: DMM device capacity (8192 Mbytes) does not match the address range assigned (512 Mbytes)
info: kernel located at: /home/ubuntu/gem5-60eb3fef9c2d/full_system_images/system/binaries/vmlinux
info: listening for system connection on port 3456
0: system.tsunami.io rtc: Real-time clock set to Thu Jan  1 00:00:00 2009
0: system.remote_gdb_listener: listening for remote gdb #0 on port 7000
Switch at instruction count:500000000
info: Entering event queue # 0. Starting simulation...
warn: Prefetch instructions in Alpha do not do anything
```

فایل خروجی با نام statsbodytrack در پوشه Q1 تمرین قرار داده شده است. مقادیر پارامترهای خواسته شده عبارتند از:

**bodytrack 2** Table

Round	Demand parameter	Simulation value
1	sim_insts	97987373
	Committed_Insts	729857
	Issued_Insts	?
	icache.overall_miss_rate	0.011687
	dcache.overall_miss_rate	0.082135
	l2.overall_miss_rate	0.592112
	cpu.dcache.read/cpu.dcache.write	133488/ 124830= 1.06
	Mem.reads/mem.writes	17835/ 8746= 2.03
2	sim_insts	48638756 2
	Committed_Insts	38765927 0
	Issued_Insts	?
	icache.overall_miss_rate	0.000661
	dcache.overall_miss_rate	0.007228
	l2.overall_miss_rate	0.513198
	cpu.dcache.read/cpu.dcache.write	69401672 /

3	sim_insts	10089516 4
	Committed_Insts	858994
	Issued_Insts	?
	icache.overall_miss_rate	0.001470
3	dcache.overall_miss_rate	0.003125
	l2.overall_miss_rate	0.476588
	cpu.dcache.read/cpu.dcache.write	163157/ 112678= 1.44
	Mem.reads/mem.writes	1028/ 353= 2.91
4	sim_insts	10190127 2
	Committed_Insts	1006108
	Issued_Insts	?
	icache.overall_miss_rate	0.016508
	dcache.overall_miss_rate	0.072920
	l2.overall_miss_rate	0.499384
	cpu.dcache.read/cpu.dcache.write	187126/ 157814= 1.18
	Mem.reads/mem.writes	21074/ 6930= 3.04

## Bodytrack .1.4

برای این برنامه نیز مشابه برنامه قبلی، از دستور زیر برای تعریف و انتخاب استفاده شده:

./writescrpts.pl bodytrack 1

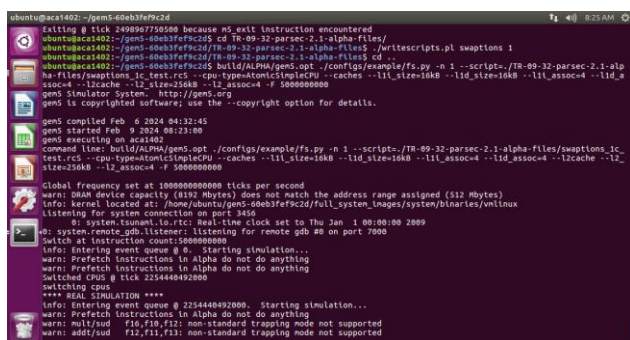
```
ubuntu@aca1402:~/gem5-60eb3fef9c2d/TR-09-32-parsec-2.1-alpha-files$ ./writescrpts.pl bodytrack 1
```

سپس gem5 را با دستور زیر برای این برنامه به صورت تک هسته اجرا می‌کنیم:

```
build/ALPHA/gem5.opt
./configs/example/fs.py -n 1
--script=./TR-09-32-parsec-2.1-alpha-files/bbodytrack_1c_test.rcs
--cpu-type=AtomicSimpleCPU
--caches
--l1i_size=16kB
--l1d_size=16kB
```

سپس gem5 را با دستور زیر برای این برنامه به صورت تک هسته اجرا می‌کنیم:

```
build/ALPHA/gem5.opt
./configs/example/fs.py -n 1
--script=./TR-09-32-parsec-2.1-alpha-
files/swaptions_1c_test.rcS
--cpu-type=AtomicSimpleCPU
--caches
--l1i_size=16kB
--l1d_size=16kB
--l1i_assoc=4
--l1d_assoc=4
--l2cache
--l2_size=256kB
--l2_assoc=4
-F 5000000000
```



فایل خروجی با نام statsswaptions در پوشه Q1 قرار داده شده است. مقادیر پارامترهای خواسته شده عبارتند از:

Table 3 Swaptions

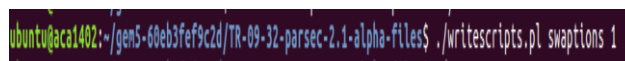
Round	Demand parameter	Simulation value
1	sim_insts	97987211
	Committed_Insts	729735
	Issued_Insts	?
	icache.overall_miss_rate	0.011687
	dcache.overall_miss_rate	0.082133
	l2.overall_miss_rate	0.592224
	cpu.dcache.read/cpu.dcache.write	133470/124819=1.06
	Mem.reads/mem.writes	17836/8745=

3		7893657=8.79
	Mem.reads/mem.writes	420971/270668=1.55
	sim_insts	584215809
	Committed_Insts	97828247
	Issued_Insts	?
	icache.overall_miss_rate	0.001336
	dcache.overall_miss_rate	0.016452
	l2.overall_miss_rate	0.591683
	cpu.dcache.read/cpu.dcache.write	21532661/5580664=3.85
	Mem.reads/mem.writes	342737/195457=1.75
4	sim_insts	585538358
	Committed_Insts	1322549
	Issued_Insts	?
	icache.overall_miss_rate	0.009503
	dcache.overall_miss_rate	0.060739
	l2.overall_miss_rate	0.553861
	cpu.dcache.read/cpu.dcache.write	235372/177393=1.32
	Mem.reads/mem.writes	21132/7062=2.99

## Swaptions .1.5

مانند دو برنامه دیگر، ابتدا از دستور زیر برای تعریف و انتخاب آن استفاده شده:

```
./writescrpts.pl swaptions 1
```





در این بخش به طور مختصر مفاهیم به کار رفته در تمرین 1 را معرفی می کنیم:

### Blackscholes

**پردازش مالی:** این برنامه برای محاسبه قیمت آپشن های اروپایی با استفاده از روش مونت کارلو طراحی شده است.

**موازی سازی:** Blackscholes برای اجرا بر روی سیستم های چند هسته ای بهینه شده است و می تواند از مزایای این سیستم ها برای افزایش سرعت محاسبات استفاده کند.

**عملکرد حافظه:** این برنامه به طور گسترده از حافظه استفاده می کند و می تواند برای تست و شبیه سازی عملکرد حافظه سیستم مورد استفاده قرار گیرد.

### Swaptions

**قیمت گذاری مشتقات:** این برنامه برای محاسبه قیمت قراردادهای مبادله نرخ بهره (Swaptions) با استفاده از روش مونت کارلو طراحی شده است.

**موازی سازی:** Swaptions مانند Blackscholes برای اجرا بر روی سیستم های چند هسته ای بهینه شده است.

**عملکرد حافظه:** این برنامه نیز به طور گسترده از حافظه استفاده می کند و می تواند برای تست و شبیه سازی عملکرد حافظه سیستم مورد استفاده قرار گیرد.

### Bodytrack

**پردازش تصویر:** این برنامه برای ردیابی حرکات بدن انسان در تصاویر و ویدئوها طراحی شده است.

**بینایی ماشین:** Bodytrack از الگوریتم های بینایی ماشین برای تشخیص و ردیابی اندام های بدن انسان استفاده می کند.

**پردازش موازی:** این برنامه می تواند برای پردازش موازی تصاویر و ویدئوها به منظور افزایش سرعت ردیابی حرکات بدن استفاده شود.

		2.03
2	sim_insts	100188659
	Committed_Insts	1460573
	Issued_Insts	?
	icache.overall_miss_rate	0.021343
	dcache.overall_miss_rate	0.052439
	l2.overall_miss_rate	0.318594
	cpu.dcache.read/cpu.dcache.write	274430/174414=1.57
	Mem.reads/mem.writes	17660/29562=0.59
3	sim_insts	100534482
	Committed_Insts	345823
	Issued_Insts	?
	icache.overall_miss_rate	0.009525
	dcache.overall_miss_rate	0.044334
	l2.overall_miss_rate	0.414162
	cpu.dcache.read/cpu.dcache.write	64968/27941=2.31
	Mem.reads/mem.writes	3099/1364=2.27
4	sim_insts	101577846
	Committed_Insts	1043364
	Issued_Insts	?
	icache.overall_miss_rate	0.011861
	dcache.overall_miss_rate	0.066017
	l2.overall_miss_rate	0.534973
	cpu.dcache.read/cpu.dcache.write	192229/159499=1.20
	Mem.reads/mem.writes	19266/6935=2.77

## 2. تمرین دوم: Branch Prediction

در این تمرین دو برنامه از بسته محک Parsec، streamcluster و rtview، در دو نوع CPU و سه حالت BranchPrediction (مجموعاً 12 حالت)، اجرا می‌کنیم.

فایل خروجی این مرحله با نام stats-Tournament-stream-DerivO3 در پوشه Q2 تمرین قرار داده شده است. پارامترهای مرتبط با branch prediction عبارت است از:

Table 4 **streamcluster-DerivO3-Tournament**

Round	Prediction parameter	Simulation value
1	Number of BP lookups	245221
	Number of conditional branches predicted	203730
	Number of conditional branches incorrect	11353
	Number of BTB lookups	169576
	Number of BTB hits	83278
	BTB Hit Percentage	49.109544
	Number of times the RAS was used to get a target	14414
	Number of incorrect RAS predictions	520
	Number of branches that fetch encountered	245221
	Number of branches that fetch has predicted taken	97692
	Number of branch fetches per cycle	0.107578
	Number of times decode resolved a branch	11632
	Number of times decode detected a branch misprediction	1348
	Number of branches that were predicted taken incorrectly	3844
	Number of branches that were predicted not taken incorrectly	10158
	Number of branch mispredicts detected at execute	14002
	Number of branches executed	119221
	The number of times a branch was mispredicted	12823
	Number of branches	98734

### 2.1 Streamcluster

ابتدا این برنامه را با دستور زیر تعریف و انتخاب کردیم:

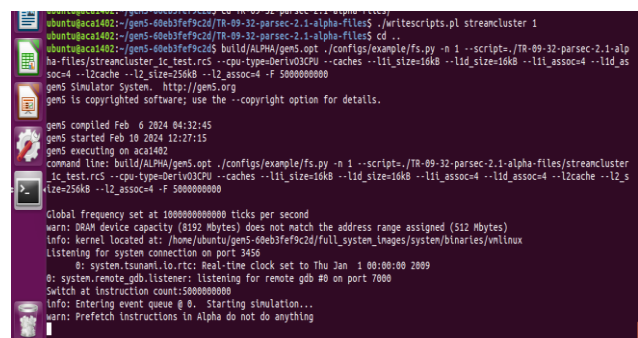
```
./writescrpts.pl streamcluster 1
```

سپس با DerivO3CPU با سه نوع BranchPrediction اجرا کردیم.

#### • DerivO3-TournamentBP

باتوجه به اینکه Predictor به صورت پیش فرض روی Tournament تنظیم بود، بدون تغییر آن و با دستور زیر برنامه را اجرا کردیم:

```
build/ALPHA/gem5.opt
./configs/example/fs.py -n 1
--script=./TR-09-32-parsec-2.1-alpha-files/streamcluster_1c_test.rcS
--cpu-type=DerivO3CPU
--caches
--l1i_size=16kB
--l1d_size=16kB
--l1i_assoc=4
--l1d_assoc=4
--l2cache
--l2_size=256kB
--l2_assoc=4
-F 5000000000
```





	Number of times the RAS was used to get a target	20861
	Number of incorrect RAS predictions	476
	Number of branches that fetch encountered	197147
	Number of branches that fetch has predicted taken	102268
	Number of branch fetches per cycle	0.102752
	Number of times decode resolved a branch	21316
	Number of times decode detected a branch misprediction	1299
	Number of branches that were predicted taken incorrectly	2742
	Number of branches that were predicted not taken incorrectly	8569
	Number of branch mis predicts detected at execute	11311
	Number of branches executed	135784
	The number of times a branch was mis-predicted	10932
	Number of branches committed	128062
4	Number of BP lookups	243326
	Number of conditional branches predicted	191862
	Number of conditional branches incorrect	11942
	Number of BTB lookups	162440
	Number of BTB hits	103462
	BTB Hit Percentage	63.692440
	Number of times the RAS was used to get a target	18791
	Number of incorrect RAS predictions	548
	Number of branches that fetch encountered	243326
4	Number of branches that fetch has predicted taken	122253
	Number of branch fetches	0.091462

	committed	
2	Number of BP lookups	360527
	Number of conditional branches predicted	271093
	Number of conditional branches incorrect	16852
	Number of BTB lookups	237836
	Number of BTB hits	142163
	BTB Hit Percentage	59.773541
	Number of times the RAS was used to get a target	35170
	Number of incorrect RAS predictions	1267
	Number of branches that fetch encountered	360527
	Number of branches that fetch has predicted taken	177333
	Number of branch fetches per cycle	0.092041
	Number of times decode resolved a branch	25439
	Number of times decode detected a branch misprediction	2565
	Number of branches that were predicted taken incorrectly	6779
	Number of branches that were predicted not taken incorrectly	14643
	Number of branch mis predicts detected at execute	21422
	Number of branches executed	220566
	The number of times a branch was mis-predicted	20147
	Number of branches committed	196426
3	Number of BP lookups	197147
	Number of conditional branches predicted	140738
	Number of conditional branches incorrect	7356
	Number of BTB lookups	130966
3	Number of BTB hits	81407
	BTB Hit Percentage	62.158881

```
--script=./TR-09-32-parsec-2.1-alpha-
files/streamcluster_1c_test.rcS
--cpu-type=DerivO3CPU
--caches
--l1i_size=16kB
--l1d_size=16kB
--l1i_assoc=4
--l1d_assoc=4
--l2cache
--l2_size=256kB
--l2_assoc=4
-F 5000000000
```

فایل خروجی این مرحله با نام

stats-Bimode-stream-DerivO3 در پوشه Q2 تمرین

قرار داده شده است.

مقادیر پارامترهای خواسته شده عبارتند از:

**Streamcluster - Derivo - BiMode5** Table

Round	Prediction parameter	Simulation value
1	Number of BP lookups	258304
	Number of conditional branches predicted	209638
	Number of conditional branches incorrect	13438
	Number of BTB lookups	99908
	Number of BTB hits	77022
	BTB Hit Percentage	77.092925
	Number of times the RAS was used to get a target	16078
	Number of incorrect RAS predictions	556
	Number of branches that fetch encountered	258304
	Number of branches that fetch has predicted taken	93100
	Number of branch fetches per cycle	0.117282
	Number of times decode resolved a branch	14648
	Number of times decode detected a branch misprediction	1433

per cycle	
Number of times decode resolved a branch	14594
Number of times decode detected a branch misprediction	1428
Number of branches that were predicted taken incorrectly	5470
Number of branches that were predicted not taken incorrectly	9312
Number of branch mis predicts detected at execute	14782
Number of branches executed	142042
The number of times a branch was mis-predicted	13729
Number of branches committed	121075

### DerivO3-BiModeBP •

در این مرحله ابتدا باید نوع Predictor را به BiModeBP تغییر دهیم. برای این کار، در دایرکتوری src/cpu/o3 استفاده از تکست ادیتور nano، در سطر مربوط به Branch Predictor، نوع پیش بینی کننده را عوض کردیم:

nano O3CPU.py

```

# flags, so we need 4-5 times the number of CC regs. ex
# physical integer regs to be sure we don't run out. In
# typical real machines, CC regs are not explicitly renamed
# i.e. it's a side effect of the reg renaming, so they should
# never be the bottleneck here.
defaultNumPhysCCRegs = Self.numPhysIntRegs * 5
numPhysCCRegs = Param.Unsigned(defaultNumPhysCCRegs,
    "Number of physical CC registers")
numIntEntries = Param.Unsigned(64, "Number of instruction queue entries")
numMDEntries = Param.Unsigned(192, "Number of reorder buffer entries")

sntNumFetchThreads = Param.Unsigned(1, "SMT Number of Fetching Threads")
sntFetchPolicy = Param.String("SingleThread", "SMT Fetch policy")
sntLsqPolicy = Param.String("Partitioned", "SMT Lsq Sharing Policy")
sntLsqThreshold = Param.Int(100, "SMT Lsq Threshold Sharing Parameter")
sntLsqPolicy = Param.String("Partitioned", "SMT Lsq Sharing Policy")
sntLsqThreshold = Param.Int(100, "SMT Lsq Threshold Sharing Parameter")
sntMDEPolicy = Param.String("Partitioned", "SMT MDE Sharing Policy")
sntMDEThreshold = Param.Int(100, "SMT MDE Threshold Sharing Parameter")
sntMDEPolicy = Param.String("RoundRobin", "SMT Commit Policy")

branchPred = Param.BranchPredictor(ElModeBP(numThreads,
    Param.numThreads),
    "Branch Predictor")
needsISO = Param.Bool(building["TARGET_ISA"] == "x86",
    "Enable ISO Memory Model")

def addCheckerCpu(self):
    if building["TARGET_ISA"] in ["arm"]:

```

سپس GEM5 را دوباره بیلد کردیم. پس از بیلد، از دستور زیر برای اجرای برنامه استفاده کردیم:

```
build/ALPHA/gem5.opt
./configs/example/fs.py -n 1
```

	executed	
	The number of times a branch was mis-predicted	18549
	Number of branches committed	196391
3	Number of BP lookups	235987
	Number of conditional branches predicted	179899
	Number of conditional branches incorrect	6726
	Number of BTB lookups	145225
	Number of BTB hits	78640
	BTB Hit Percentage	54.150456
	Number of times the RAS was used to get a target	20642
	Number of incorrect RAS predictions	481
	Number of branches that fetch encountered	235987
	Number of branches that fetch has predicted taken	99282
	Number of branch fetches per cycle	0.123059
	Number of times decode resolved a branch	21341
	Number of times decode detected a branch misprediction	1297
	Number of branches that were predicted taken incorrectly	1974
	Number of branches that were predicted not taken incorrectly	8758
	Number of branch mis predicts detected at execute	10732
	Number of branches executed	135842
	The number of times a branch was mis-predicted	10308
	Number of branches committed	128064
4	Number of BP lookups	278202
4	Number of conditional branches predicted	229796

	Number of branches that were predicted taken incorrectly	2517
	Number of branches that were predicted not taken incorrectly	14174
	Number of branch mis predicts detected at execute	16691
	Number of branches executed	123579
	The number of times a branch was mis-predicted	14832
	Number of branches committed	98734
2	Number of BP lookups	430825
	Number of conditional branches predicted	341594
	Number of conditional branches incorrect	15247
	Number of BTB lookups	233800
	Number of BTB hits	134142
	BTB Hit Percentage	57.374679
	Number of times the RAS was used to get a target	34785
	Number of incorrect RAS predictions	1310
	Number of branches that fetch encountered	430825
	Number of branches that fetch has predicted taken	168927
	Number of branch fetches per cycle	0.110331
	Number of times decode resolved a branch	25505
	Number of times decode detected a branch misprediction	2556
	Number of branches that were predicted taken incorrectly	4619
	Number of branches that were predicted not taken incorrectly	15166
	Number of branch mis predicts detected at execute	19785
	Number of branches	219751

```

ubuntu@aca1402: ~/gem5-dsdb3f9c2d/src/cpu/03
GNU nano 2.5.3 File: 03CPU.py
# flags, to us need 4-5 times the number of CC regs as
# physical integer regs to be sure we don't run out. In
# typical real machines, CC regs are not explicitly renamed
# (it's a side effect of int reg renaming), so they should
# never be the bottleneck here.
defaultNumPhysCRegs = Self.numPhysIntRegs * 5
numPhysCRegs = Param.Unsigned(defaultNumPhysCRegs,
                                "Number of physical CC registers")
numInstEntries = Param.Unsigned(64, "Number of instruction queue entries")
numReorderEntries = Param.Unsigned(192, "Number of reorder buffer entries")

smtNumFetchThreads = Param.Unsigned(1, "SMT Number of Fetching Threads")
smtFetchPolicy = Param.String("SingleThread", "SMT Fetch policy")
smtSQPolicy = Param.String("Partitioned", "SMT SQ Sharing Policy")
smtSQThreshold = Param.Int(100, "SMT SQ Threshold Sharing Parameter")
smtSQPolicy = Param.String("Partitioned", "SMT SQ Sharing Policy")
smtIQThreshold = Param.Int(100, "SMT IQ Threshold Sharing Parameter")
smtROBPolicy = Param.String("Partitioned", "SMT ROB Sharing Policy")
smtROBThreshold = Param.Int(100, "SMT ROB Threshold Sharing Parameter")
smtCommitPolicy = Param.String("RoundRobin", "SMT Commit Policy")

branchPred = Param.BranchPredictor(TAGE(numThreads, Parent.numThreads),
                                     "Branch Predictor")
needsTSD = Param.Bool(buildEnv["TARGET_ISA"] == "x86",
                      "Enable TSD Memory Model")

def addCheckerCpu(self):
    if buildEnv["TARGET_ISA"] in ["arm"]:

```

اما برای بیلد مجدد GEM5 با خطای زیر مواجه شدیم.

```

ubuntu@aca1402: ~/gem5-dsdb3f9c2d
scons build/ALPHA/gem5.opt -j4scons: Reading SconsScript files ...
Checking for leading underscore in global variables... (cached) no
Checking for C header file python.h... (cached) yes
Checking for C library python2.7... (cached) yes
Checking for C library pthread... (cached) yes
Checking for C library dl... (cached) yes
Checking for C library util... (cached) yes
Checking for C library m... (cached) yes
Checking for accept(0,0,0) in C++ library None... (cached) yes
Checking for __libc_version() in C++ library 2... (cached) yes
Checking for __GLIBC__ in C++ library 2... (cached) yes
Checking for clock_gettime(CLOCK_MONOTONIC, NULL, NULL) in C++ library None... (cached) yes
Checking for timer_create(CLOCK_MONOTONIC, NULL, NULL) in C++ library None... (cached) no
Checking for timer_create(CLOCK_MONOTONIC, NULL, NULL) in C++ library rt... (cached) yes
Checking for C library tcmalloc... (cached) yes
Checking for C header file fenv.h... (cached) yes
Checking for C header file linux/kvm.h... (cached) yes
Checking size of struct kvm_xsave... (cached) yes
Checking for member exclude_host in struct perf_event_attr... (cached) yes
Building in /home/ubuntu/gem5-dsdb3f9c2d/build/ALPHA
Using saved variables file /home/ubuntu/gem5-dsdb3f9c2d/build/variables/ALPHA
NameError: name 'TAGE' is not defined
File /home/ubuntu/gem5-dsdb3f9c2d/SconsScript, line 1405:
    SconsScript('src/SconsScript', variant_dir = variant_path, exports = 'env')
File /usr/lib/scons/Scons/Script/SconsScript.py, line 614:
    return method(args, **kw)
File /usr/lib/scons/Scons/Script/SconsScript.py, line 551:
    return SconsScript(self.fs, *files, **subst_kw)
File /usr/lib/scons/Scons/Script/SconsScript.py, line 260:
    exec file in call_stack[-1].globals

```

باتوجه به اینکه این Predictor در این ورژن قابل اجرا نبود، از LocalBP استفاده کردیم.

### DerivO3- LocalBP

مشابه مراحل قبل، برای تغییر نوع Predictor و سپس بیلد GEM5 بعد از اعمال تغییر، اقدام کردیم.

```

ubuntu@aca1402: ~/gem5-dsdb3f9c2d/src/cpu/03
GNU nano 2.5.3 File: 03CPU.py
# flags, to us need 4-5 times the number of CC regs as
# physical integer regs to be sure we don't run out. In
# typical real machines, CC regs are not explicitly renamed
# (it's a side effect of int reg renaming), so they should
# never be the bottleneck here.
defaultNumPhysCRegs = Self.numPhysIntRegs * 5
numPhysCRegs = Param.Unsigned(defaultNumPhysCRegs,
                                "Number of physical CC registers")
numInstEntries = Param.Unsigned(64, "Number of instruction queue entries")
numReorderEntries = Param.Unsigned(192, "Number of reorder buffer entries")

smtNumFetchThreads = Param.Unsigned(1, "SMT Number of Fetching Threads")
smtFetchPolicy = Param.String("SingleThread", "SMT Fetch policy")
smtSQPolicy = Param.String("Partitioned", "SMT SQ Sharing Policy")
smtSQThreshold = Param.Int(100, "SMT SQ Threshold Sharing Parameter")
smtSQPolicy = Param.String("Partitioned", "SMT SQ Sharing Policy")
smtIQThreshold = Param.Int(100, "SMT IQ Threshold Sharing Parameter")
smtROBPolicy = Param.String("Partitioned", "SMT ROB Sharing Policy")
smtROBThreshold = Param.Int(100, "SMT ROB Threshold Sharing Parameter")
smtCommitPolicy = Param.String("RoundRobin", "SMT Commit Policy")

branchPred = Param.BranchPredictor(LocalBP(numThreads, Parent.numThreads),
                                     "Branch Predictor")
needsTSD = Param.Bool(buildEnv["TARGET_ISA"] == "x86",
                      "Enable TSD Memory Model")

def addCheckerCpu(self):
    if buildEnv["TARGET_ISA"] in ["arm"]:

```

### DerivO3- TAGE

Number of conditional branches incorrect	10090
Number of BTB lookups	182030
Number of BTB hits	94709
BTB Hit Percentage	52.029336
Number of times the RAS was used to get a target	18124
Number of incorrect RAS predictions	532
Number of branches that fetch encountered	278202
Number of branches that fetch has predicted taken	112833
Number of branch fetches per cycle	0.104330
Number of times decode resolved a branch	13571
Number of times decode detected a branch misprediction	1367
Number of branches that were predicted taken incorrectly	3844
Number of branches that were predicted not taken incorrectly	9068
Number of branch mis predicts detected at execute	12912
Number of branches executed	140141
The number of times a branch was mis-predicted	11929
Number of branches committed	121075

ابتدا مشابه مراحل قبل، Branch Predictor را به TAGE تغییر دادیم.

نتایج خروجی در فایلی به نام stats-LocalBP-stream-DerivO3 در پوشه Q2 تمرین قرار داده شده است. همچنین مقادیر پارامترهای خوسته شده عبارتند از:

Streamcluster - Derivo - LocalBP6 Table

Round	Prediction parameter	Simulation
-------	----------------------	------------

	Number of BTB hits	148487
	BTB Hit Percentage	52.567352
	Number of times the RAS was used to get a target	34490
	Number of incorrect RAS predictions	1248
	Number of branches that fetch encountered	356522
	Number of branches that fetch has predicted taken	182977
	Number of branch fetches per cycle	0.091153
	Number of times decode resolved a branch	24593
	Number of times decode detected a branch misprediction	2448
	Number of branches that were predicted taken incorrectly	7898
	Number of branches that were predicted not taken incorrectly	14236
	Number of branch mis predicts detected at execute	22134
	Number of branches executed	220796
	The number of times a branch was mis-predicted	20701
	Number of branches committed	196070
3	Number of BP lookups	204194
	Number of conditional branches predicted	147292
	Number of conditional branches incorrect	8471
	Number of BTB lookups	167189
	Number of BTB hits	90103
	BTB Hit Percentage	53.892900
	Number of times the RAS was used to get a target	21263
	Number of incorrect RAS predictions	469

		value
1	Number of BP lookups	244758
	Number of conditional branches predicted	205638
	Number of conditional branches incorrect	10407
	Number of BTB lookups	172147
	Number of BTB hits	86251
	BTB Hit Percentage	50.103110
	Number of times the RAS was used to get a target	13631
	Number of incorrect RAS predictions	503
	Number of branches that fetch encountered	244758
	Number of branches that fetch has predicted taken	99882
	Number of branch fetches per cycle	0.107644
	Number of times decode resolved a branch	10990
	Number of times decode detected a branch misprediction	1297
	Number of branches that were predicted taken incorrectly	3997
	Number of branches that were predicted not taken incorrectly	8956
	Number of branch mis predicts detected at execute	12953
	Number of branches executed	117548
	The number of times a branch was mis-predicted	11916
	Number of branches committed	98734
2	Number of BP lookups	356522
	Number of conditional branches predicted	269030
	Number of conditional branches incorrect	17302
	Number of BTB lookups	282470



Number of times decode detected a branch misprediction	1362
Number of branches that were predicted taken incorrectly	4999
Number of branches that were predicted not taken incorrectly	9449
Number of branch mis predicts detected at execute	14448
Number of branches executed	141321
The number of times a branch was mis-predicted	13326
Number of branches committed	121077

#### MinorCPU •

پردازنده نوع MinorCPU با دستور شامل فرکانس، ارور زیر را می‌دهد:

#### Error: Segmentation Fault (Core Dumped)

```

dun@aca1402:~/gem5-680b3fe9c2d$ build/ALPHA/gem5.opt ./configs/example/fs.py -n 1 --script=./TR-09-32-parsec-2.1-alpha-files/streamcluster_1c_test.rcs --cpu-type=MinorCPU --caches --l1i_size=16kB --l1d_size=16kB --l1i_assoc=4 --l1d_esso
c=4 --l2cache --l2_size=256kB --l2_assoc=4 -F 5000000000
gem5 Simulator System, http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 compiled Feb 10 2024 13:39:05
gem5 started Feb 10 2024 13:46:41
gem5 executing on aca1402
command line: build/ALPHA/gem5.opt ./configs/example/fs.py -n 1 --script=./TR-09-32-parsec-2.1-alpha-files/streamcluster_1c_test.rcs --cpu-type=MinorCPU --caches --l1i_size=16kB --l1d_size=16kB --l1i_assoc=4 --l1d_esso
c=4 --l2cache --l2_size=256kB --l2_assoc=4 -F 5000000000

Global frequency set at 1000000000000 ticks per second
warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (512 Mbytes)
info: kernel located at: /home/dun@/gem5-680b3fe9c2d/full_system_images/system/binaries/vmlinux
listening for system connection on port 3456
0: system.tsunami.io.rtc: Real-time clock set to Thu Jan 1 00:00:00 2009
0: system.remote.gdb.listener: listening for remote gdb #0 on port 7000
Switch at instruction count:5000000000
info: Entering event queue @ 0. Starting simulation...
Segmentation Fault (core dumped)

```

ولی با دستور زیر (بدون تعریف فرکانس) اجرا شد:

```

build/ALPHA/gem5.opt
./configs/example/fs.py -n 1
--script=./TR-09-32-parsec-2.1-alpha-
files/streamcluster_1c_test.rcs
--cpu-type=MinorCPU
--caches
--l1i_size=16kB
--l1d_size=16kB

```

Number of branches that fetch encountered	204194
Number of branches that fetch has predicted taken	111366
Number of branch fetches per cycle	0.105129
Number of times decode resolved a branch	21399
Number of times decode detected a branch misprediction	1267
Number of branches that were predicted taken incorrectly	4001
Number of branches that were predicted not taken incorrectly	8521
Number of branch mis predicts detected at execute	12522
Number of branches executed	137916
The number of times a branch was mis-predicted	12084
Number of branches committed	128189
Number of BP lookups	245641
Number of conditional branches predicted	196534
Number of conditional branches incorrect	11475
Number of BTB lookups	198879
Number of BTB hits	103579
BTB Hit Percentage	52.081416
Number of times the RAS was used to get a target	18237
Number of incorrect RAS predictions	532
Number of branches that fetch encountered	245641
Number of branches that fetch has predicted taken	121816
Number of branch fetches per cycle	0.091821
Number of times decode resolved a branch	13524



Number of conditional branches predicted	20356251
Number of conditional branches incorrect	569567
Number of BTB lookups	17046454
Number of BTB hits	9873221
BTB Hit Percentage	57.919500
Number of times the RAS was used to get a target	1579286
Number of incorrect RAS predictions	40554

```
--l1i_assoc=4
--l1d_assoc=4
--l2cache
--l2_size=256kB
--l2_assoc=4
```

## Minor-BiMode •

در این مرحله ابتدا باید نوع Predictor را به BiModeBP تغییر دهیم. برای این کار، در دایرکتوری src/cpu/minor و با استفاده از تکست ادیتور nano، در سطر مربوط به Branch Predictor، نوع پیش بینی کننده را عوض کردیم:

nano MinorCPU.py

سپس GEM5 را دوباره بیلد کردیم. پس از بیلد، از دستور زیر برای اجرای برنامه استفاده کردیم:

```
build/ALPHA/gem5.opt
./configs/example/fs.py -n 1
--script=./TR-09-32-parsec-2.1-alpha-
files/streamcluster_1c_test.rcS
--cpu-type=MinorCPU
--caches
--l1i_size=16kB
--l1d_size=16kB
--l1i_assoc=4
--l1d_assoc=4
--l2cache
```

## Minor - Tournament •

باتوجه به اینکه Predictor به صورت پیش فرض روی Tournament تنظیم بود، بدون تغییر آن و با دستور زیر برنامه را اجرا کردیم:

```
build/ALPHA/gem5.opt
./configs/example/fs.py -n 1
--script=./TR-09-32-parsec-2.1-alpha-
files/streamcluster_1c_test.rcS
--cpu-type=DerivO3CPU
--caches
--l1i_size=16kB
--l1d_size=16kB
--l1i_assoc=4
--l1d_assoc=4
--l2cache
--l2_size=256kB
--l2_assoc=4
```

فایل خروجی این مرحله با نام

stats-Tournament-stream-Minor Q2 تمرین

قرار داده شده است.

پارامترهای مرتبط با branch prediction عبارت است از:

streamcluster-Minor-Tournament7 Table

Prediction parameter	Simulation value
Number of BP lookups	24075875

--l2\_size=256kB

--l2\_assoc=4

فایل خروجی این مرحله با نام

stats-Bimode-stream-minor در پوشه Q2 تمرین قرار

داده شده است.

مقادیر پارامترهای خواسته شده عبارتند از:

**Streamcluster - Minor - BiMode8** Table

Prediction parameter	Simulation value
Number of BP lookups	24115083
Number of conditional branches predicted	20413869
Number of conditional branches incorrect	525713
Number of BTB lookups	12238555
Number of BTB hits	9725825
BTB Hit Percentage	79.468736
Number of times the RAS was used to get a target	1576705
Number of incorrect RAS predictions	43584

هم چنین مقادیر پارامترهای مربوط به Branch Prediction در جدول زیر آورده شده است:

Table 9 **rtview- Derivo - Tournament**

Round	Prediction parameter	Simulation value
1	Number of BP lookups	244849
	Number of conditional branches predicted	203398
	Number of conditional branches incorrect	11261
	Number of BTB lookups	172484
	Number of BTB hits	83437
	BTB Hit Percentage	48.373762
	Number of times the RAS was used to get a target	14387
	Number of incorrect RAS predictions	517
	Number of branches that fetch encountered	244849
	Number of branches that fetch has predicted taken	97824
	Number of branch fetches per cycle	0.111068
	Number of times decode resolved a branch	11586
	Number of times decode detected a branch misprediction	1345
	Number of branches that were predicted taken incorrectly	3789
	Number of branches that were predicted not taken incorrectly	10114
	Number of branch mispredicts detected at execute	13903
	Number of branches executed	119377
	The number of times a branch was mis-predicted	12729
	Number of branches committed	98742
2	Number of BP lookups	845876

## 2.2. rtview

ابتدا این برنامه را با دستور زیر تعریف و انتخاب کردیم:

```
./writescrpts.pl rtview 1
```

سپس با پردازنده DerivO3CPU و با سه نوع BranchPrediction اجرا کردیم.

### • DerivO3- TournamentBP

باتوجه به اینکه Predictor به صورت پیش فرض روی Tournament تنظیم بود، بدون تغییر آن و با دستور زیر برنامه را اجرا کردیم:

```
build/ALPHA/gem5.opt
./configs/example/fs.py -n 1
--script=./TR-09-32-parsec-2.1-alpha-
files/rtview_1c_test.rcS
--cpu-type=DerivO3CPU
--caches
--l1i_size=16kB
--l1d_size=16kB
--l1i_assoc=4
--l1d_assoc=4
--l2cache
--l2_size=256kB
--l2_assoc=4
-F 5000000000
```

فایل خروجی این مرحله با نام

stats-Tournament-rtview-DerivO3 در پوشه Q2

تمرین قرار داده شده است.

	Number of incorrect RAS predictions	75
	Number of branches that fetch encountered	38869
	Number of branches that fetch has predicted taken	25344
	Number of branch fetches per cycle	0.028463
	Number of times decode resolved a branch	1840
	Number of times decode detected a branch misprediction	169
	Number of branches that were predicted taken incorrectly	977
	Number of branches that were predicted not taken incorrectly	1219
	Number of branch mis predicts detected at execute	2196
	Number of branches executed	26498
	The number of times a branch was mis-predicted	2120
	Number of branches committed	23767
4	Number of BP lookups	246840
	Number of conditional branches predicted	194205
	Number of conditional branches incorrect	12001
	Number of BTB lookups	164571
	Number of BTB hits	103545
	BTB Hit Percentage	62.918133
	Number of times the RAS was used to get a target	19321
	Number of incorrect RAS predictions	582
	Number of branches that fetch encountered	246840
	Number of branches that fetch has predicted taken	122866
	Number of branch fetches per cycle	0.089647
	Number of times decode	14662

	Number of conditional branches predicted	598526
	Number of conditional branches incorrect	37524
	Number of BTB lookups	585994
	Number of BTB hits	342707
	BTB Hit Percentage	58.483022
	Number of times the RAS was used to get a target	100288
	Number of incorrect RAS predictions	2867
	Number of branches that fetch encountered	845876
	Number of branches that fetch has predicted taken	442995
	Number of branch fetches per cycle	0.089590
	Number of times decode resolved a branch	76065
	Number of times decode detected a branch misprediction	5790
	Number of branches that were predicted taken incorrectly	15216
	Number of branches that were predicted not taken incorrectly	33672
	Number of branch mis predicts detected at execute	48888
	Number of branches executed	551551
	The number of times a branch was mis-predicted	46240
	Number of branches committed	501405
3	Number of BP lookups	38869
	Number of conditional branches predicted	30810
	Number of conditional branches incorrect	1927
	Number of BTB lookups	27658
	Number of BTB hits	22237
3	BTB Hit Percentage	80.399884
	Number of times the RAS was used to get a target	3107

```
--l1d_size=16kB
--l1i_assoc=4
--l1d_assoc=4
--l2cache
--l2_size=256kB
--l2_assoc=4
-F 5000000000
```

خروجی این مرحله با نام

stats-Bimode-rtview-DerivO3 در پوشه Q2 تمرین

قرار دارد.

مقادیر پارامترهای مطلوب در جدول زیر آورده شده است:

Round	Prediction parameter	Simulation value
1	Number of BP lookups	258391
	Number of conditional branches predicted	209740
	Number of conditional branches incorrect	13461
	Number of BTB lookups	99911
	Number of BTB hits	77053
	BTB Hit Percentage	77.121638
	Number of times the RAS was used to get a target	16063
	Number of incorrect RAS predictions	558
	Number of branches that fetch encountered	258391
	Number of branches that fetch has predicted taken	93116
	Number of branch fetches per cycle	0.117245
	Number of times decode resolved a branch	14673
	Number of times decode detected a branch misprediction	1434
	Number of branches that were predicted taken incorrectly	2531
	Number of branches that	14177

resolved a branch	
Number of times decode detected a branch misprediction	1481
Number of branches that were predicted taken incorrectly	5325
Number of branches that were predicted not taken incorrectly	9678
Number of branch mispredicts detected at execute	15003
Number of branches executed	144819
The number of times a branch was mis-predicted	13846
Number of branches committed	124877

### DerivO3- BiModeBP •

در این مرحله ابتدا باید نوع Predictor را به BiModeBP تغییر دهیم. برای این کار، در دایرکتوری src/cpu/o3 و با استفاده از تکست ادیتور nano، در سطر مربوط به Branch Predictor، نوع پیش بینی کننده را عوض کردیم:

nano O3CPU.py

بعد مجدداً GEM5 را بیلد کرده و پس از بیلد موفق، دستور زیر را برای اجرای برنامه اجرا می کنیم:

```
build/ALPHA/gem5.opt
./configs/example/fs.py -n 1
--script=./TR-09-32-parsec-2.1-alpha-
files/rtview_1c_test.rcS
--cpu-type=DerivO3CPU
--caches
--l1i_size=16kB
```

	committed	
3	Number of BP lookups	43276
	Number of conditional branches predicted	35711
	Number of conditional branches incorrect	1741
	Number of BTB lookups	29797
	Number of BTB hits	21134
	BTB Hit Percentage	70.926603
	Number of times the RAS was used to get a target	2998
	Number of incorrect RAS predictions	79
	Number of branches that fetch encountered	43276
	Number of branches that fetch has predicted taken	24132
	Number of branch fetches per cycle	0.031734
	Number of times decode resolved a branch	1605
	Number of times decode detected a branch misprediction	161
	Number of branches that were predicted taken incorrectly	706
	Number of branches that were predicted not taken incorrectly	1311
	Number of branch mispredicts detected at execute	2017
	Number of branches executed	26245
	The number of times a branch was mis-predicted	1944
	Number of branches committed	23767
4	Number of BP lookups	285624
	Number of conditional branches predicted	235478
	Number of conditional branches incorrect	10337
	Number of BTB lookups	186824
	Number of BTB hits	97690
	BTB Hit Percentage	52.289856

	were predicted not taken incorrectly	
	Number of branch mispredicts detected at execute	16708
	Number of branches executed	123683
	The number of times a branch was mis-predicted	14853
	Number of branches committed	98742
2	Number of BP lookups	1088283
	Number of conditional branches predicted	843133
	Number of conditional branches incorrect	32943
	Number of BTB lookups	608489
	Number of BTB hits	322773
	BTB Hit Percentage	53.045002
	Number of times the RAS was used to get a target	98943
	Number of incorrect RAS predictions	2898
	Number of branches that fetch encountered	1088283
	Number of branches that fetch has predicted taken	421716
	Number of branch fetches per cycle	0.115822
	Number of times decode resolved a branch	74916
	Number of times decode detected a branch misprediction	5700
	Number of branches that were predicted taken incorrectly	10143
	Number of branches that were predicted not taken incorrectly	34121
	Number of branch mispredicts detected at execute	44264
	Number of branches executed	549608
	The number of times a branch was mis-predicted	41736
	Number of branches	501580



```

ubuntu@kali402:~/gem5-60eb3fe9c2d$ scons build/ALPHA/gem5-opt -j4scons: Reading Sconscript files ...
Checking for leading underscore in global variables... (cached) no
Checking for C header file python.h... (cached) yes
Checking for C library pthread... (cached) yes
Checking for C library dl... (cached) yes
Checking for C library util... (cached) yes
Checking for C library n... (cached) yes
Checking for accept(0,0,0) in C++ library None... (cached) yes
Checking for __libversion__ in C++ library... (cached) yes
Checking for GOOGLE_PROTOBUF_VERIFY_VERSION in C++ library protobuf... (cached) yes
Checking for clock_gettime(0,0,NULL,NULL) in C library None... (cached) yes
Checking for timer_create(CLOCK_MONOTONIC, NULL, NULL) in C library None... (cached) no
Checking for timer_create(CLOCK_MONOTONIC, NULL, NULL) in C library rt... (cached) yes
Checking for C library localtime... (cached) yes
Checking for C header file fenv.h... (cached) yes
Checking for C header file linux/kvm.h... (cached) yes
Checking size of struct kvm_xsave... (cached) yes
Checking for member exclude_host in struct perf_event_attr... (cached) yes
Building in /home/ubuntu/gem5-60eb3fe9c2d/build/ALPHA
Using saved variables file /home/ubuntu/gem5-60eb3fe9c2d/build/variables/ALPHA
NameError: name 'TAGE' is not defined
File /home/ubuntu/gem5-60eb3fe9c2d/SConstruct, line 1405:
SConstruct('src/SConstruct', variant_dir = variant_path, exports = 'env')
File /usr/lib/scons/Script/SConstruct.py, line 614:
return method(*args, **kw)
File /usr/lib/scons/Script/SConstruct.py, line 551:
return SConstruct(self.fs, *files, **subst_kw)
File /usr/lib/scons/Script/SConstruct.py, line 260:
exec_file = call_stack[-1].globals

```

باتوجه به اینکه این Predictor در این ورژن قابل اجرا نبود، از LocalBP استفاده کردیم.

### DerivO3- LocalBP •

مشابه مراحل قبل، برای تغییر نوع Predictor و سپس بیلد GEM5 بعد از اعمال تغییر، اقدام کردیم.

نتایج خروجی در فایلی به نام

stats-LocalBP-rtview-DerivO3 در پوشه Q2

تمرین قرار داده شده است.

مقادیر پارامترهای مربوط به Branch Predictor نیز در

جدول زیر آورده شده است:

Round	Prediction parameter	Simulation value
1	Number of BP lookups	244961
	Number of conditional branches predicted	205751
	Number of conditional branches incorrect	10439
	Number of BTB lookups	172668
	Number of BTB hits	86274
	BTB Hit Percentage	49.965251

Number of times the RAS was used to get a target	18665
Number of incorrect RAS predictions	571
Number of branches that fetch encountered	285624
Number of branches that fetch has predicted taken	116355
Number of branch fetches per cycle	0.104067
Number of times decode resolved a branch	14196
Number of times decode detected a branch misprediction	1429
Number of branches that were predicted taken incorrectly	3943
Number of branches that were predicted not taken incorrectly	9309
Number of branch mispredicts detected at execute	13252
Number of branches executed	144103
The number of times a branch was mis-predicted	12219
Number of branches committed	124875

### DerivO3- TAGE •

ابتدا مشابه مراحل قبل، Branch Predictor را به TAGE

تغییر دادیم.

اما برای بیلد مجدد GEM5 با خطای زیر مواجه شدیم:

```

build/ALPHA/gem5.opt
./configs/example/fs.py -n 1
--script=./TR-09-32-parsec-2.1-alpha-
files/rtview_1c_test.rcS
--cpu-type=MinorCPU
--caches
--l1i_size=16kB
--l1d_size=16kB
--l1i_assoc=4
--l1d_assoc=4
--l2cache
--l2_size=256kB
--l2_assoc=4

```

## Minor - Tournament •

باتوجه به اینکه Predictor به صورت پیش فرض روی Tournament تنظیم بود، بدون تغییر آن و با دستور زیر برنامه را اجرا کردیم:

```

build/ALPHA/gem5.opt
./configs/example/fs.py -n 1
--script=./TR-09-32-parsec-2.1-alpha-
files/rtview_1c_test.rcS
--cpu-type=DerivO3CPU
--caches
--l1i_size=16kB
--l1d_size=16kB
--l1i_assoc=4
--l1d_assoc=4
--l2cache
--l2_size=256kB
--l2_assoc=4

```

فایل خروجی این مرحله با نام

Number of times the RAS was used to get a target	13660
Number of incorrect RAS predictions	511
Number of branches that fetch encountered	244961
Number of branches that fetch has predicted taken	99934
Number of branch fetches per cycle	0.107734
Number of times decode resolved a branch	10996
Number of times decode detected a branch misprediction	1303
Number of branches that were predicted taken incorrectly	4019
Number of branches that were predicted not taken incorrectly	8951
Number of branch mis predicts detected at execute	12970
Number of branches executed	117564
The number of times a branch was mis-predicted	11945
Number of branches committed	501388

## MinorCPU •

پردازنده نوع MinorCPU با دستور شامل فرکانس، ارور زیر را می دهد:

Error: Segmentation Fault (Core Dumped)

ولی با دستور زیر (بدون تعریف فرکانس) اجرا شد:



### 3. جمع بندی

ابتدا مفاهیم زیر را که در تمرین استفاده شده است، به طور مختصر تعریف می کنیم:

#### ❖ پیش بینی کننده های شاخه:

- Tournament از یک الگوریتم پوبا برای انتخاب بهترین پیش بینی بین دو پیش بینی کننده دیگر استفاده می کند.
- LocalBP از یک تاریخچه محلی از نتایج شاخه برای پیش بینی آینده استفاده می کند.
- BiModeBP از دو پیش بینی کننده جداگانه برای پیش بینی شاخه های شرطی و غیرشرطی استفاده می کند.
- TAGE: یک پیش بینی کننده برنج است که برای پیش بینی دقیق تر مسیرهای شرطی در برنامه های کامپیوتری طراحی شده است. TAGE از دو جدول برای پیش بینی مسیرهای شرطی استفاده می کند:
- Local History: این جدول اطلاعات مربوط به شاخه های اخیر در یک بلوک کد را ذخیره می کند.
- Global Prediction این جدول اطلاعات مربوط به شاخه ها در کل برنامه را ذخیره می کند.

TAGE برای پیش بینی مسیر یک برنج، ابتدا به جدول تاریخچه محلی مراجعه می کند. اگر اطلاعات مربوط به آن شاخه در جدول تاریخچه محلی موجود باشد، TAGE از آن اطلاعات برای پیش بینی مسیر شاخه استفاده می کند. در غیر این صورت، TAGE به جدول پیش بینی جهانی مراجعه می کند.

تفاوت بین پنج مارک های **rtview** و **streamcluster** در بسته **Parsec**:

**rtview**:

- پنج مارکی برای سنجش عملکرد پردازنده در پردازش تصاویر است.

- بر روی عملیات پردازش تصویر مانند فیلتر کردن، تبدیل و ترکیب تصاویر تمرکز دارد.
- از تصاویر با وضوح بالا و مجموعه داده های بزرگ استفاده می کند.
- برای سیستم هایی با حافظه نهان بزرگ و پهنای باند حافظه بالا مناسب است.

#### **Streamcluster**:

- پنج مارکی برای سنجش عملکرد پردازنده در خوشه بندی داده ها است.
- بر روی الگوریتم های خوشه بندی مبتنی بر جریان تمرکز دارد.
- از مجموعه داده های بزرگ با ابعاد بالا استفاده می کند.
- برای سیستم هایی با حافظه نهان بزرگ و قدرت پردازش موازی بالا مناسب است.

#### تفاوت های کلیدی:

نوع عملیات: **rtview** بر روی عملیات پردازش تصویر تمرکز دارد، در حالی که **streamcluster** بر روی الگوریتم های خوشه بندی داده ها تمرکز دارد.

نوع داده: **rtview** از تصاویر با وضوح بالا و مجموعه داده های بزرگ استفاده می کند، در حالی که **streamcluster** از مجموعه داده های بزرگ با ابعاد بالا استفاده می کند.

نیازمندی های سیستم: **rtview** برای سیستم هایی با حافظه نهان بزرگ و پهنای باند حافظه بالا مناسب است، در حالی که **streamcluster** برای سیستم هایی با حافظه نهان بزرگ و قدرت پردازش موازی بالا مناسب است.

#### ❖ انواع پردازنده:

**AtomicSimpleCPU** یک پردازنده ساده و ابتدایی است که از یک مدل محاسباتی ساده برای اجرای دستورالعمل ها استفاده می کند. این پردازنده از ویژگی های پیشرفته ای مانند پیش بینی شاخه، بازیابی اطلاعات از حافظه، و اجرای موازی دستورالعمل ها پشتیبانی نمی کند.

DerivO3CPU یک پردازنده پیچیده تر است که از یک مدل محاسباتی پیشرفته برای اجرای دستورالعمل ها استفاده می کند. این پردازنده از ویژگی های پیشرفته ای مانند پیش بینی شاخه، بازیابی اطلاعات از حافظه، و اجرای موازی دستورالعمل ها پشتیبانی می کند.

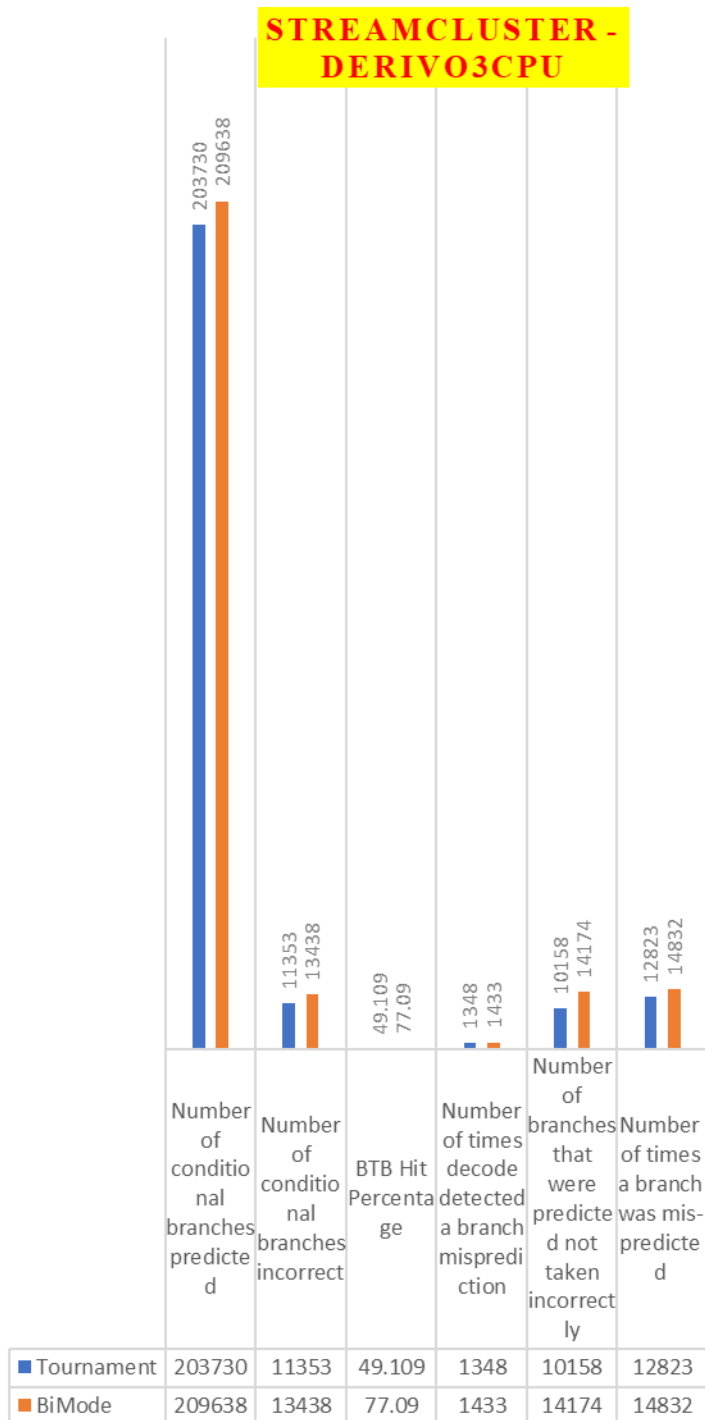
در DerivO3CPU، از پیش بینی شاخه استفاده می شود. این بدان معناست که پردازنده می تواند شاخه ها را پیش بینی کند و فقط دستورالعمل های پس از شاخه هایی را که اجرا می شوند، اجرا کند. این امر می تواند باعث افزایش IPC در DerivO3CPU شود.

MinorCPU یک پردازنده کمکی است که برای وظایف کم اهمیت و سبک در سیستم های کامپیوتری استفاده می شود. این پردازنده وظایف کم اهمیت را از پردازنده اصلی (CPU) جدا می کند و به طور موازی با آن اجرا می کند. این امر باعث می شود که CPU اصلی برای وظایف مهم تر و سنگین تر آزاد شود و در نتیجه عملکرد کلی سیستم افزایش یابد.

در ادامه مقایسه نموداری از نتایج تمرین 2 آورده می شود.

### ❖ مقایسه نتایج پیش بینی کننده‌ها

ابتدا نتایج پیش بینی کننده های Tournament و BiMode را برای برنامه streamcluster با هم مقایسه می کنیم.  
(برای مقایسه در نمودار زیر، فقط برخی پارامترها از راند 1 هر جدول پیش بینی کننده Tournament و BiMode آورده شده است).

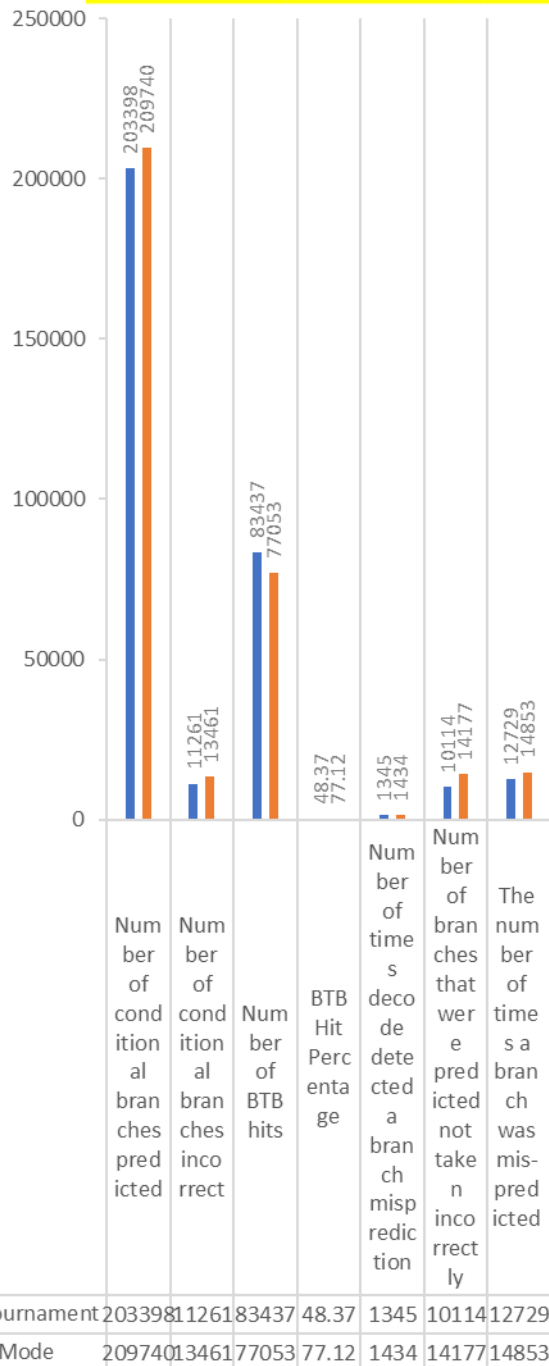


نمودار بالا، نشان می دهد که پیش بینی کننده Tournament عملکرد بهتر و تعداد خطای کمتری را نسبت به BiMode در این شبیه سازی به دست آورده است.

یکی از دلایل اصلی بهبود عملکرد TournamentBP نسبت به BiModeBP می تواند این باشد که الگوریتم Tournament



## RTVIEW - DERIVO3CPU



به صورت پویا نوع پیش بینی را انتخاب می کند. یعنی عملکرد دو روش Local و Global را پروفایل کرده و هر کدام که بهتر باشد را انتخاب می کند.

در این بخش نتایج پیش بینی کننده های Tournament و BiMode را برای برنامه rtview با هم مقایسه می کنیم. (برای مقایسه در نمودار زیر، فقط برخی پارامترها از راند 1 هر جدول پیش بینی کننده Tournament و BiMode آورده شده است).

نمودار بالا نشان می دهد که پیش بینی کننده Tournament خطای کمتر و عملکرد بهتری نسبت به BiMode داشته است.