EE/CE/CS 6304 Computer Architecture Gem5

Benjamin Carrion Schaefer
Assistant Professor
Department of Electrical and Computer Engineering

Overview

- Gem5 resources

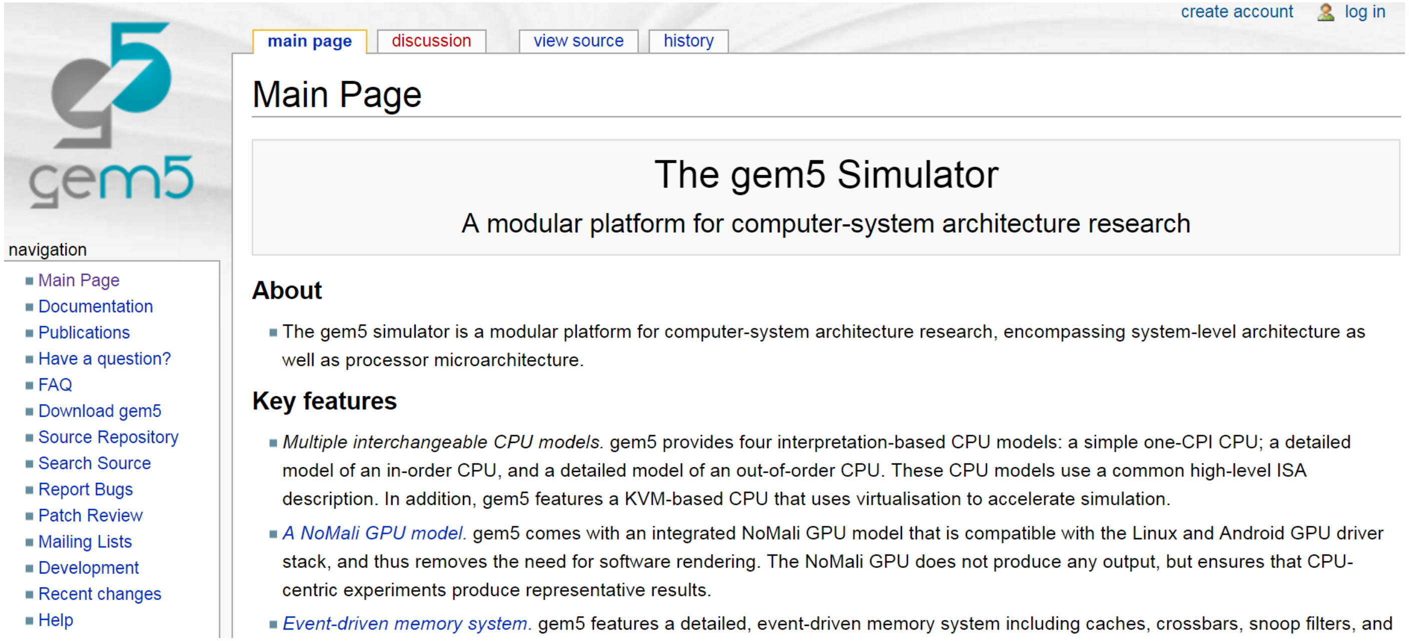
1. Gem5 introduction
2. login to CE6304 Server system
3. Download the Benchmarks
4. Run the experiment for 429.mcf benchmark

Learning about gem5

- http://learning.gem5.org/
- https://github.com/powerjg/learning_gem5
- Google.com

gem5

- Ref : http://gem5.org/Main_Page



Main Page

The gem5 Simulator

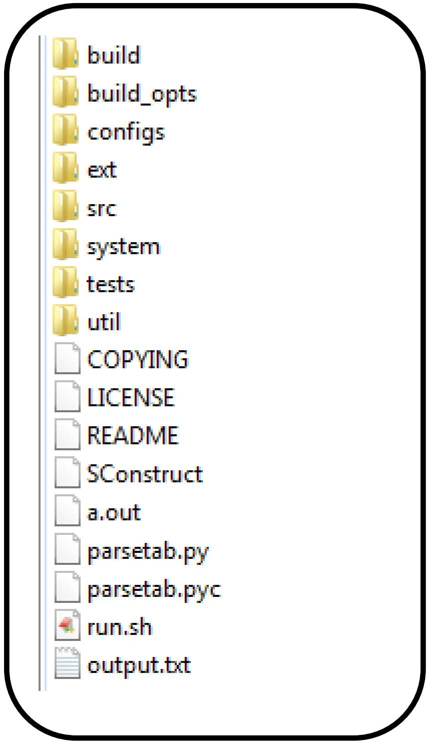
A modular platform for computer-system architecture research

About

- The gem5 simulator is a modular platform for computer-system architecture research, encompassing system-level architecture as well as processor microarchitecture.

Key features

- *Multiple interchangeable CPU models.* gem5 provides four interpretation-based CPU models: a simple one-CPI CPU; a detailed model of an in-order CPU, and a detailed model of an out-of-order CPU. These CPU models use a common high-level ISA description. In addition, gem5 features a KVM-based CPU that uses virtualisation to accelerate simulation.
- *A NoMali GPU model.* gem5 comes with an integrated NoMali GPU model that is compatible with the Linux and Android GPU driver stack, and thus removes the need for software rendering. The NoMali GPU does not produce any output, but ensures that CPU-centric experiments produce representative results.
- *Event-driven memory system.* gem5 features a detailed, event-driven memory system including caches, crossbars, snoop filters, and



```
build
build_opts
configs
ext
src
system
tests
util
COPYING
LICENSE
README
SConstruct
a.out
parsetab.py
parsetab.pyc
run.sh
output.txt
```

1. gem5

- Modular platform for computer-system architecture research, encompassing system-level architecture as well as processor microarchitecture
- Four CPU models
  - Simple (one CPI) CPU
  - Detailed in-order CPU
  - Detailed out-of-order CPI
  - In order pipelined model
- All CPU models use a common high-level ISA

1. gem5

- gem5 is written primarily in C++ and python
- It can simulate a complete system:
  - With devices and an operating system in **full system mode (FS mode)**
  - User space only programs where system services are provided directly by the simulator in **syscall emulation mode (SE mode)**.
- There are varying levels of support for executing Alpha, ARM, MIPS, Power, SPARC, and 64-bit x86 binaries on any of the 4 CPU models

Download and Install

```
%git clone https://gem5.googlesource.com/public/gem5
%cd gem5
%scons build/X86/gem5.opt –j5
```

Git= Online repository . Can be downloaded directly from Linux terminal

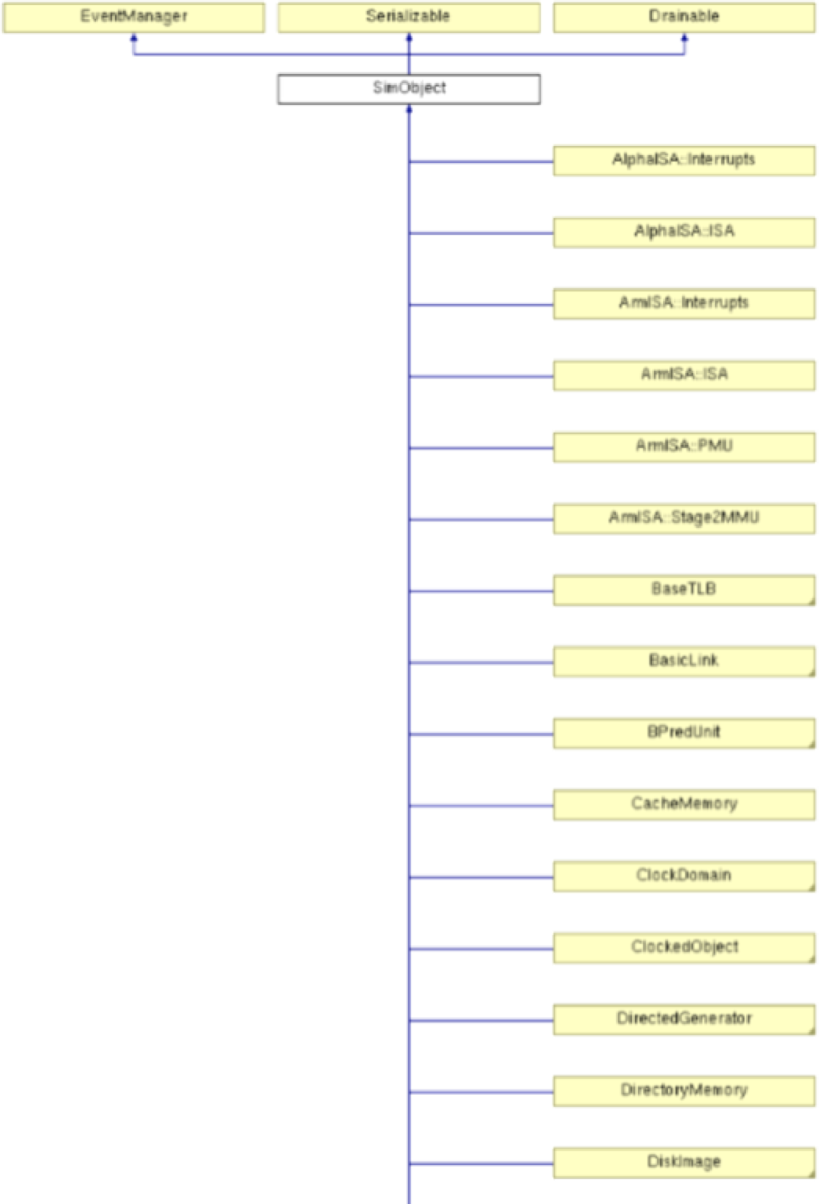
cd = change directory

Scons = Builds the system (similar to make) – See http://scons.org

build/X86/gem5.opt = Parameter passed to scons

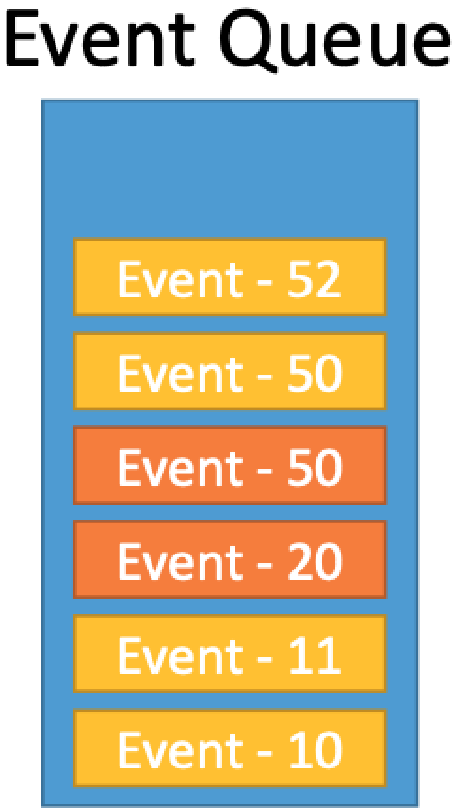
Gem5 Architecture

- gem5 consists of “**SimObjects**” Most C++ objects in gem5 inherit
- from **class SimObject**
- Represent physical system components



Gem5 Simulator

- Gem5 is a discrete event simulator
  1. Event is added to queue
  2. Event is dequeued
  3. Event is executed



Event Queue

Event - 52

Event - 50

Event - 50

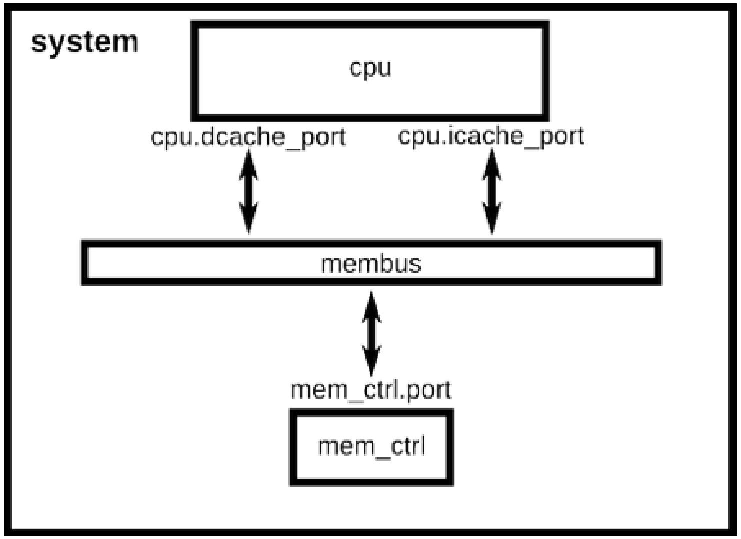
Event - 20

Event - 11

Event - 10

Gem5 configuration scripts

- http://learning.gem5.org/book/part1/simple_config.html http://learning.gem5.org/book/part1/cache_config.html
- **gem5 binary takes, as a parameter, a <u>python</u> script**
  - Sets up and executes the simulation
  - Creates all of the components of the system
  - Specifies all of the parameters for the system components

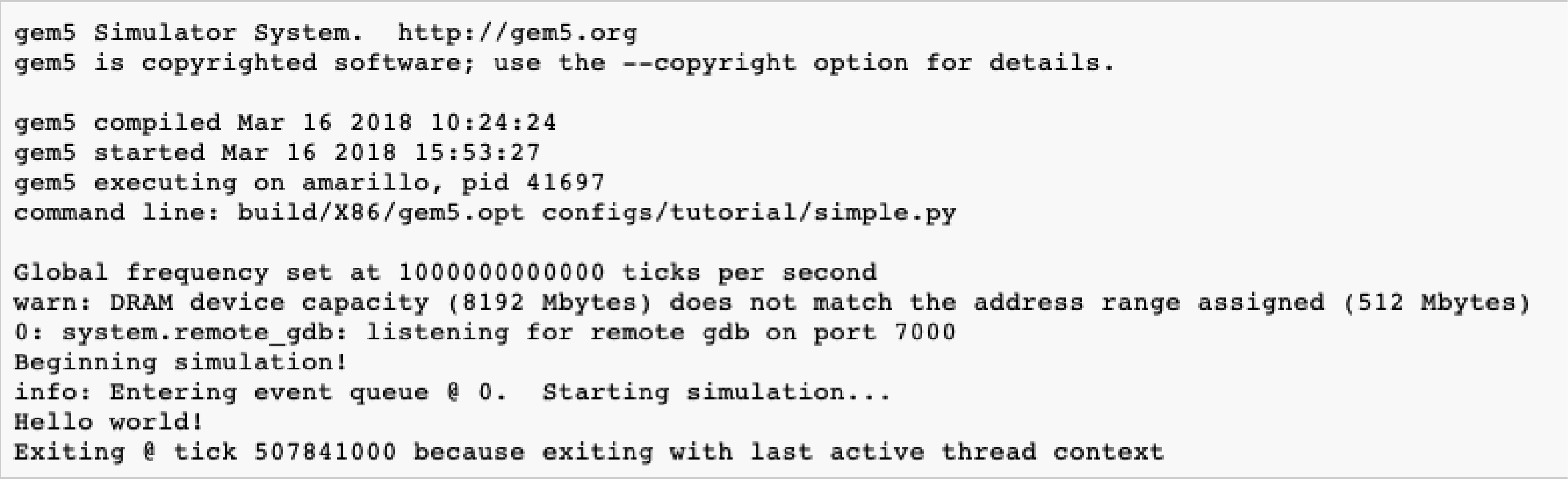


http://learning.gem5.org/book/part1/simple_config.html

Running gem5

%build/X86/gem5.opt configs/tutorial/simple.py

Output:

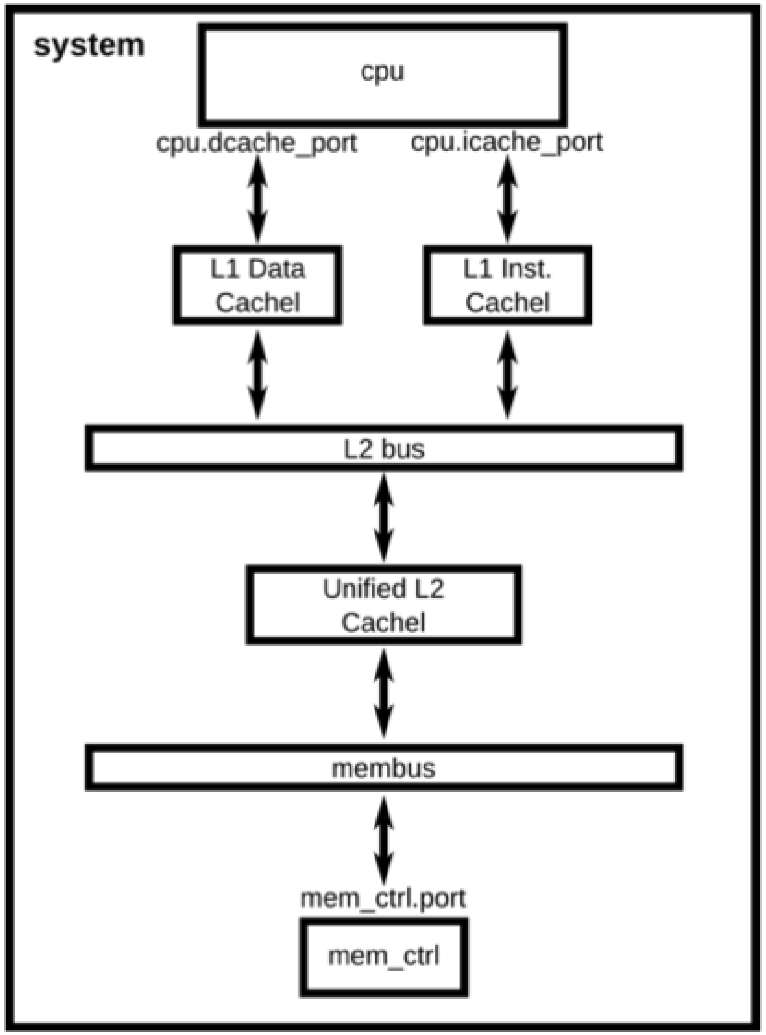


```
gem5 Simulator System.  http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 compiled Mar 16 2018 10:24:24
gem5 started Mar 16 2018 15:53:27
gem5 executing on amarillo, pid 41697
command line: build/X86/gem5.opt configs/tutorial/simple.py

Global frequency set at 1000000000000 ticks per second
warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (512 Mbytes)
0: system.remote_gdb: listening for remote gdb on port 7000
Beginning simulation!
info: Entering event queue @ 0.  Starting simulation...
Hello world!
Exiting @ tick 507841000 because exiting with last active thread context
```

Results change if configuration parameters are modified, e.g., type of CPU or type of DDR controller

Caches in gem5

- http://learning.gem5.org/book/part1/cache_config.html
- Extending SimObjects in Python
- Object-oriented



Gem5 Output

http://learning.gem5.org/book/part1/gem5_stats.html

%ls m5out

Filename	Description
Config.ini	Dumps all of the parameters Simobjects Shows exactly what you simulated
Config.json	Same as config.ini, but in json format (JavaScript Object Notation)
Stats.txt	Detailed statistic output. Each SimObject defines and updates statistics. They are printed here at the end of simulation

gem5 - Usage

- The gem5 command line has four parts,
  1. the gem5 binary
  2. options for the binary
  3. a python simulation script
  4. Options for the script
- The options that are passed to the gem5 binary and those passed to the script are handled separately, so be sure any options you use are being passed to the right component.

```
% <gem5 binary> [gem5 options] <simulation script> [script options]
```

gem5 - Usage –con’t

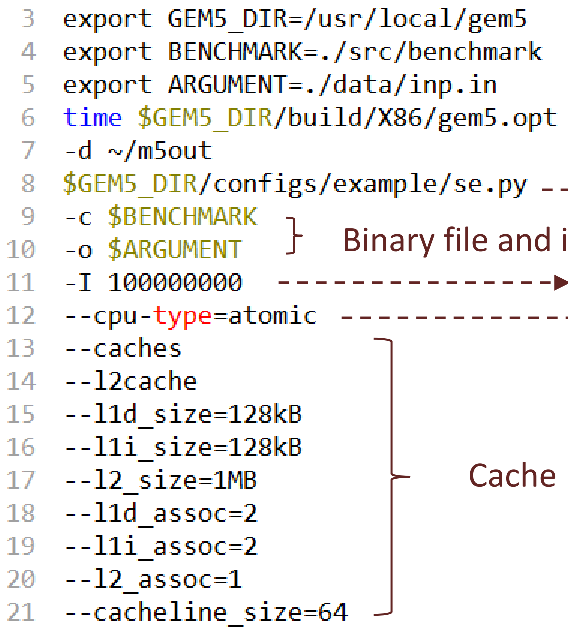
command line: build/ALPHA/gem5.opt configs/example/se.py -h

```
Usage: se.py [options]


Options:
  -h, --help          show this help message and exit
  -c CMD, --cmd=CMD     The binary to run in syscall emulation mode.
  -o OPTIONS, --options=OPTIONS ,
      The options to pass to the binary, use " " around the entire string
  -i INPUT, --input=INPUT Read stdin from a file.
  --output=OUTPUT       Redirect stdout to a file.
  --errout=ERROUT       Redirect stderr to a file.
  -d, --detailed
  -t, --timing
  --inorder
```

```
-n NUM_CPUS, --num-cpus=NUM_CPUS
 --caches
 --l2cache
 --fastmem
 --clock=CLOCK
 --num-dirs=NUM_DIRS
 --num-l2caches=NUM_L2CACHES
 --l1d_size=L1D_SIZE
 --l2_size=L2_SIZE
 --l1d_assoc=L1D_ASSOC
 --l2_assoc=L2_ASSOC
…
```

gem5 – Example

% <gem5 binary> [gem5 options] <simulation script> [script options]



```
export GEM5_DIR=/usr/local/gem5
export BENCHMARK=./src/benchmark
export ARGUMENT=./data/inp.in
time $GEM5_DIR/build/X86/gem5.opt
-d ~/m5out
$GEM5_DIR/configs/example/se.py
-c $BENCHMARK
-o $ARGUMENT
-I 100000000
--cpu-type=atomic
--caches
--l2cache
--l1d_size=128kB
--l1i_size=128kB
--l2_size=1MB
--l1d_assoc=2
--l1i_assoc=2
--l2_assoc=1
--cacheline_size=64
```

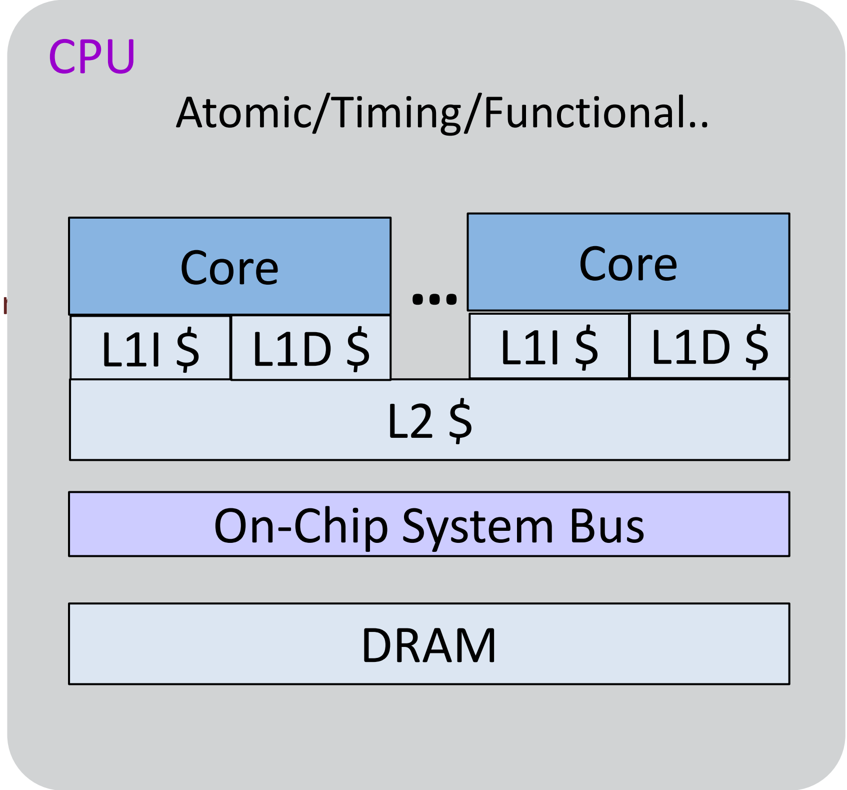
System call emulation mode

Binary file and its input

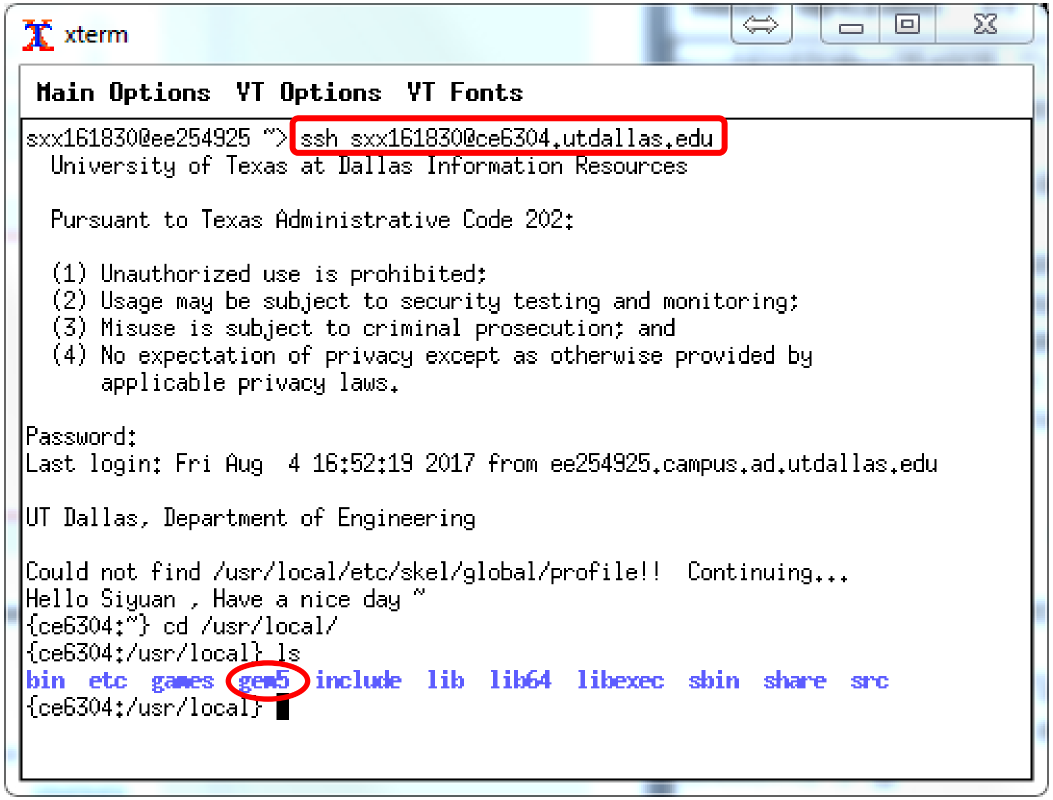
Simulation Instruction Numbe

Atomic CPU

Cache Info



2. Login to Server system



```
sxx161830@ee254925 ~> ssh sxx161830@ce6304.utdallas.edu
  University of Texas at Dallas Information Resources

  Pursuant to Texas Administrative Code 202:

  (1) Unauthorized use is prohibited;
  (2) Usage may be subject to security testing and monitoring;
  (3) Misuse is subject to criminal prosecution; and
  (4) No expectation of privacy except as otherwise provided by
      applicable privacy laws.

Password:
Last login: Fri Aug  4 16:52:19 2017 from ee254925.campus.ad.utdallas.edu

UT Dallas, Department of Engineering

Could not find /usr/local/etc/skel/global/profile!!  Continuing...
Hello Siyuan , Have a nice day ^
{ce6304:~} cd /usr/local/
{ce6304:/usr/local} ls
bin  etc  games  gem5  include  lib  lib64  libexec  sbin  share  src
{ce6304:/usr/local}
```

Login to Server system:

- ○ Your netid @ ce6304.utdallas.edu
- ○ Your password(the same as galaxy)

Off-campus ,you need to setup the VPN, info: http://www.utdallas.edu/oit/vpn/

OIT Home » Cisco AnyConnect VPN

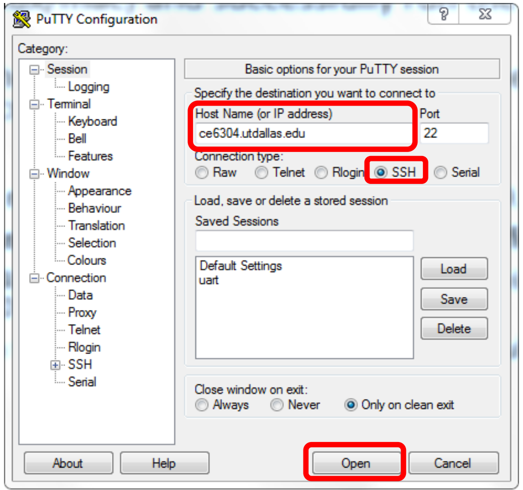
Cisco AnyConnect VPN

- ➢ Gem5 is installed in the /usr/local/ folder
- ➢ **NOTE:** You **DO NOT** have write permissions on this folder → work on your home directory

2. Login to Server system -- Con’t

- Log on server from Windows:
- 1. Open PuTTY, and find PuTTY configuration window;
- 2. Set host name to be “ce6304.utdallas.edu”;
- 3. Select connection type to be “SSH”;
- 4. Click on open;
- 5. Fills in your NetID and password in coming windows.

- Log on server from Mac/Linux:
- 1. Open Terminal;
- 2. Type in command “ssh <your-net-id>@ce6304.utdallas.edu”;
- 3. Type in your password.



Specify the destination you want to connect to

Host Name (or IP address): ce6304.utdallas.edu

Port: 22

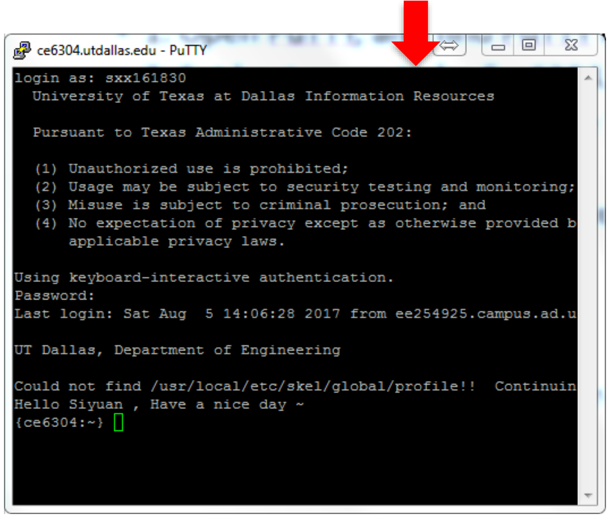
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings
uart

Close window on exit: ○ Always ○ Never ● Only on clean exit



```
login as: sxx161830
  University of Texas at Dallas Information Resources

  Pursuant to Texas Administrative Code 202:

  (1) Unauthorized use is prohibited;
  (2) Usage may be subject to security testing and monitoring;
  (3) Misuse is subject to criminal prosecution; and
  (4) No expectation of privacy except as otherwise provided b
      applicable privacy laws.

Using keyboard-interactive authentication.
Password:
Last login: Sat Aug  5 14:06:28 2017 from ee254925.campus.ad.u

UT Dallas, Department of Engineering

Could not find /usr/local/etc/skel/global/profile!!  Continuin
Hello Siyuan , Have a nice day ~
{ce6304:~}
```

2. Login to Server system -- Con’t

- If you are using your own Linux machine you can install the gem5 as you can modify it by yourself (download link):

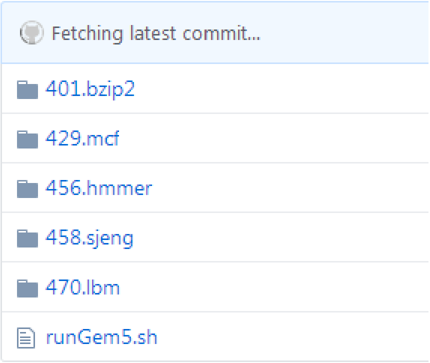
Steps to install gem5 are generally broken down into two steps:

- ➢ first, install all dependencies; second, install gem5 itself. Detailed steps can be found in following links.
- Install dependencies: http://gem5.org/Dependencies
- Install gem5 itself: http://gem5.org/Build_System
- ➢ Particularly, if you choose to install gem5 on Ubuntu system, you have an easier way to install dependencies (sudo for access administrator access):

➢sudo apt-get install python-dev scons m4 build-essential g++ swig zlib-dev

3.Download the Benchmarks

- We will use a set of CPU benchmarks to simulate your architectural design. You can download the benchmarks at the following link:
- ➢ https://github.com/timberjack/Project1_SPEC.git
- Or you could download the zip file from the following link:
- ➢ https://github.com/timberjack/Project1_SPEC



Fetching latest commit...

401.bzip2

429.mcf

456.hmmer

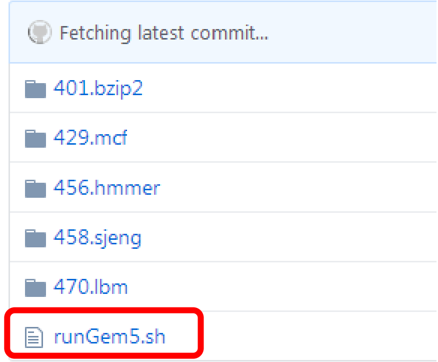
458.sjeng

470.lbm

runGem5.sh

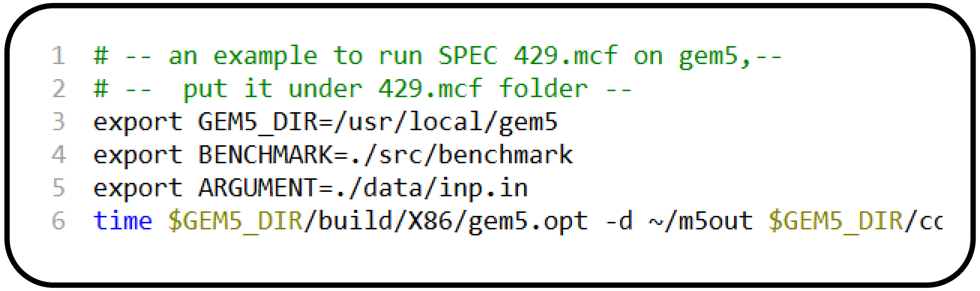
4. Run the experiments for 429.mcf benchmark

- Within “**spec**” folder, you can also find the script “**runGem5.sh**”. It is a sample script that runs gem5 on benchmark **429.mcf.** You can use this as an example how to specify the command line to run a benchmark program on gem5.



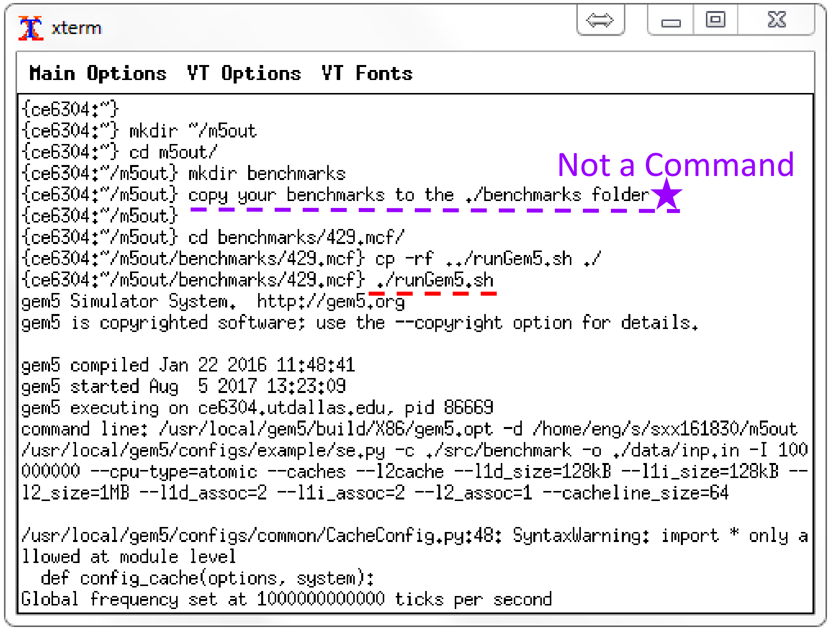
Fetching latest commit...

- 401.bzip2
- 429.mcf
- 456.hmmer
- 458.sjeng
- 470.lbm
- runGem5.sh



```
# -- an example to run SPEC 429.mcf on gem5,--
# --  put it under 429.mcf folder --
export GEM5_DIR=/usr/local/gem5
export BENCHMARK=./src/benchmark
export ARGUMENT=./data/inp.in
time $GEM5_DIR/build/X86/gem5.opt -d ~/m5out $GEM5_DIR/cc
```

4. Run the experiments for 429.mcf benchmark

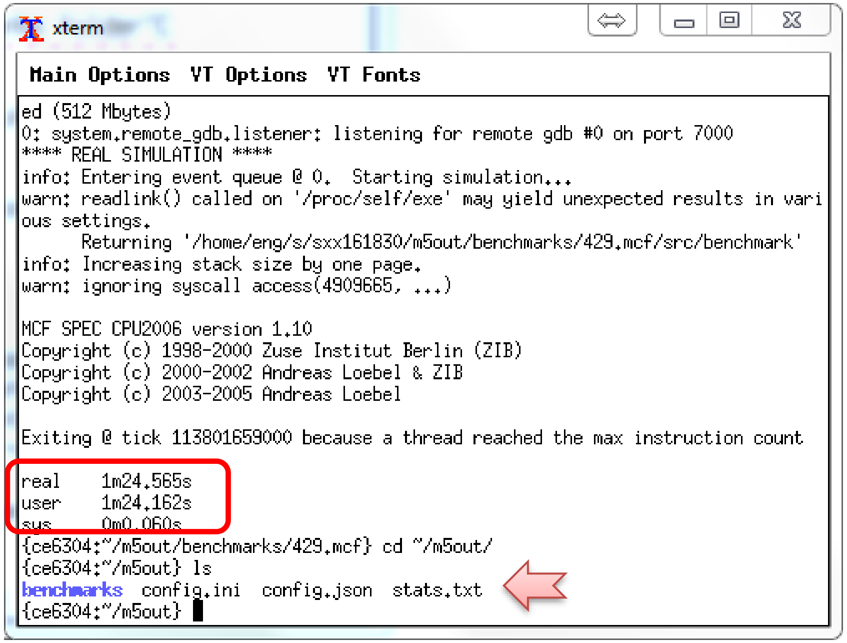


```
{ce6304:~}
{ce6304:~} mkdir ~/m5out
{ce6304:~} cd m5out/
{ce6304:~/m5out} mkdir benchmarks
{ce6304:~/m5out} copy your benchmarks to the ./benchmarks folder
{ce6304:~/m5out}
{ce6304:~/m5out} cd benchmarks/429.mcf/
{ce6304:~/m5out/benchmarks/429.mcf} cp -rf ../runGem5.sh ./
{ce6304:~/m5out/benchmarks/429.mcf} ./runGem5.sh
gem5 Simulator System.  http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 compiled Jan 22 2016 11:48:41
gem5 started Aug  5 2017 13:23:09
gem5 executing on ce6304.utdallas.edu, pid 86669
command line: /usr/local/gem5/build/X86/gem5.opt -d /home/eng/s/sxx161830/m5out
/usr/local/gem5/configs/example/se.py -c ./src/benchmark -o ./data/inp.in -I 100
000000 --cpu-type=atomic --caches --l2cache --l1d_size=128kB --l1i_size=128kB --
l2_size=1MB --l1d_assoc=2 --l1i_assoc=2 --l2_assoc=1 --cacheline_size=64

/usr/local/gem5/configs/common/CacheConfig.py:48: SyntaxWarning: import * only a
llowed at module level
  def config_cache(options, system):
Global frequency set at 1000000000000 ticks per second
```

Not a Command ★

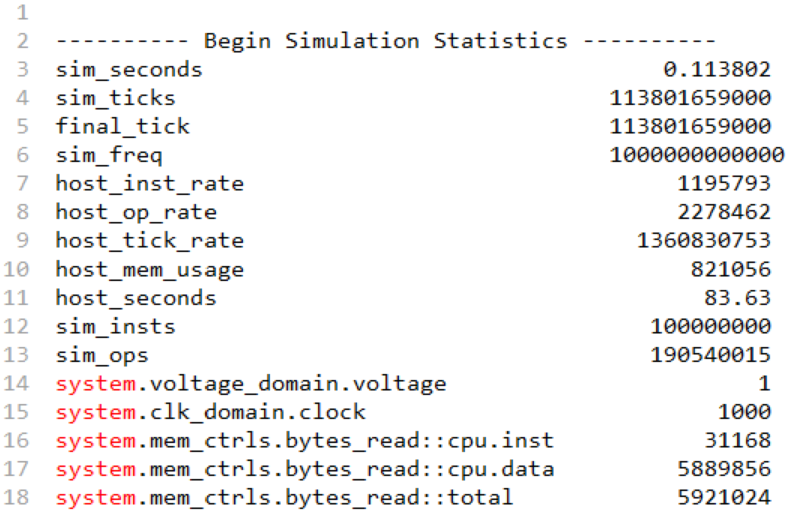
1. Create the folder ~/m5out
2. Go to that folder, then create the folder ./benchmarks
3. Copy the spec(downloads) to the ./benchmarks folder
4. To run SPEC 429.mcf on gem5, put ./runGem5.sh under 429.mcf folder
5. Run the ./runGem5.sh

4. Run the experiments for 429.mcf benchmark



```
ed (512 Mbytes)
0: system.remote_gdb.listener: listening for remote gdb #0 on port 7000
**** REAL SIMULATION ****
info: Entering event queue @ 0.  Starting simulation...
warn: readlink() called on '/proc/self/exe' may yield unexpected results in vari
ous settings.
      Returning '/home/eng/s/sxx161830/m5out/benchmarks/429.mcf/src/benchmark'
info: Increasing stack size by one page.
warn: ignoring syscall access(4909665, ...)

MCF SPEC CPU2006 version 1.10
Copyright (c) 1998-2000 Zuse Institut Berlin (ZIB)
Copyright (c) 2000-2002 Andreas Loebel & ZIB
Copyright (c) 2003-2005 Andreas Loebel

Exiting @ tick 113801659000 because a thread reached the max instruction count

real    1m24.565s
user    1m24.162s
sys     0m0.060s
{ce6304:~/m5out/benchmarks/429.mcf} cd ~/m5out/
{ce6304:~/m5out} ls
benchmarks  config.ini  config.json  stats.txt
{ce6304:~/m5out}
```

The execution of this command provides the output file “**stats.txt**” under folder “**m5out**” (default, or into your defined directory).



```

---------- Begin Simulation Statistics ----------
sim_seconds                                  0.113802
sim_ticks                                113801659000
final_tick                               113801659000
sim_freq                                 1000000000000
host_inst_rate                                1195793
host_op_rate                                  2278462
host_tick_rate                             1360830753
host_mem_usage                                 821056
host_seconds                                    83.63
sim_insts                                   100000000
sim_ops                                     190540015
system.voltage_domain.voltage                       1
system.clk_domain.clock                          1000
system.mem_ctrls.bytes_read::cpu.inst           31168
system.mem_ctrls.bytes_read::cpu.data         5889856
system.mem_ctrls.bytes_read::total            5921024
```

Project

- Using GEM5, in this project, we will study the effect of varying Branch Prediction parameters on different benchmarks.

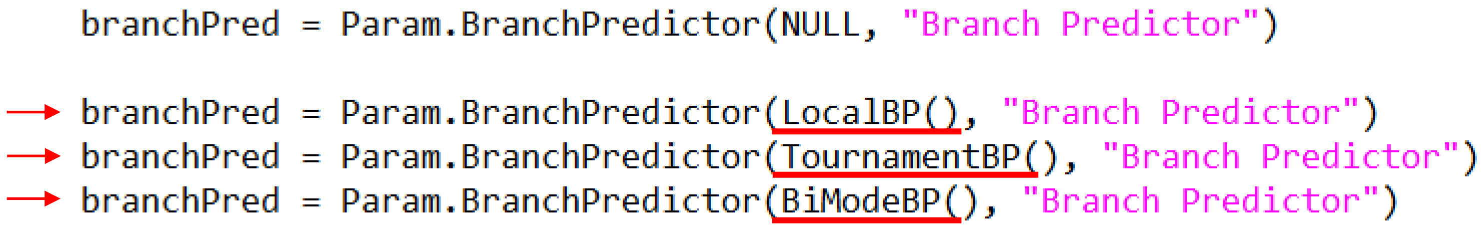
- There are 3 kinds of Branch Predictors available in GEM5 (stable).
  1. 2bit_local Predictor
  2. Bi_mode Predictor
  3. Tournament Predictor
- Groups of 2
- 10min ppt report summarizing all the results
- Grading rubric attached at the end of pdf report

Project : Branch Predictor

1. Set up the simulation environment
2. Adding Branch Predictor for TimingSimpleCPU
3. Adding extra resulting parameter in stats.txt file
4. Comparison among different branch predictors

2. Adding Branch Predictor

- By default, the Gem5 TimingSimple CPU does not have the Branch Predictor Support, thus, in the **$gem5/src/cpu/simple/BaseSimpleCPU.py**

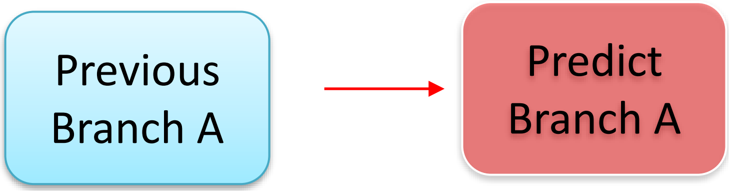


```
branchPred = Param.BranchPredictor(NULL, "Branch Predictor")

branchPred = Param.BranchPredictor(LocalBP(), "Branch Predictor")
branchPred = Param.BranchPredictor(TournamentBP(), "Branch Predictor")
branchPred = Param.BranchPredictor(BiModeBP(), "Branch Predictor")
```

LocalBP	2bit Local Branch Predictor
BiModeBP	Global Behavioral Branch Predictor
Tournament	Combined/Hybrid Predictor

2. Adding Branch Predictor

LocalBP

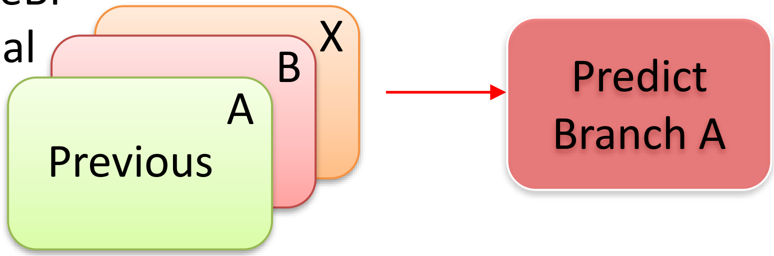


Previous Branch A

Predict Branch A

```
class LocalBP(BranchPredictor):
    type = 'LocalBP'
    cxx_class = 'LocalBP'
    cxx_header = "cpu/pred/2bit_local.hh"

    localPredictorSize = Param.Unsigned(2048, "Size of local predictor")
    localCtrBits = Param.Unsigned(2, "Bits per counter")
```

BiModeBP
Global



X

B

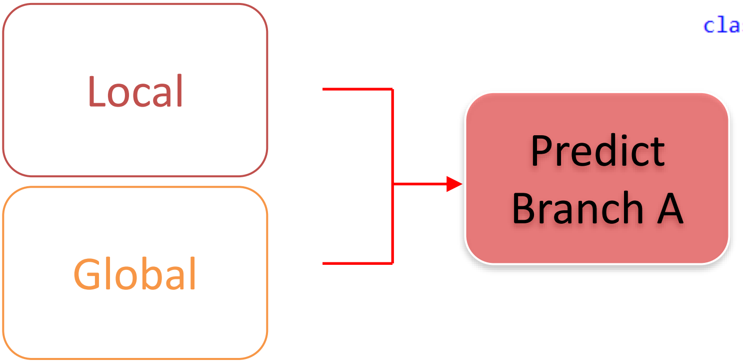
A

Previous

Predict Branch A

```
class BiModeBP(BranchPredictor):
    type = 'BiModeBP'
    cxx_class = 'BiModeBP'
    cxx_header = "cpu/pred/bi_mode.hh"

    globalPredictorSize = Param.Unsigned(8192, "Size of global predictor")
    globalCtrBits = Param.Unsigned(2, "Bits per counter")
    choicePredictorSize = Param.Unsigned(8192, "Size of choice predictor")
    choiceCtrBits = Param.Unsigned(2, "Bits of choice counters")
```

Tournament
Hybrid



Local

Global

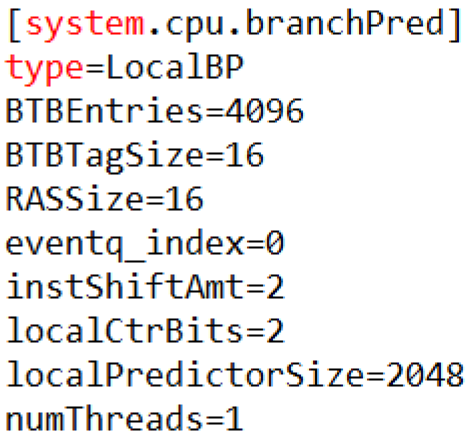
Predict Branch A

```
class TournamentBP(BranchPredictor):
    type = 'TournamentBP'
    cxx_class = 'TournamentBP'
    cxx_header = "cpu/pred/tournament.hh"

    localPredictorSize = Param.Unsigned(2048, "Size of local predictor")
    localCtrBits = Param.Unsigned(2, "Bits per counter")
    localHistoryTableSize = Param.Unsigned(2048, "size of local history table")
    globalPredictorSize = Param.Unsigned(8192, "Size of global predictor")
    globalCtrBits = Param.Unsigned(2, "Bits per counter")
    choicePredictorSize = Param.Unsigned(8192, "Size of choice predictor")
    choiceCtrBits = Param.Unsigned(2, "Bits of choice counters")
```

2. Adding Branch Predictor

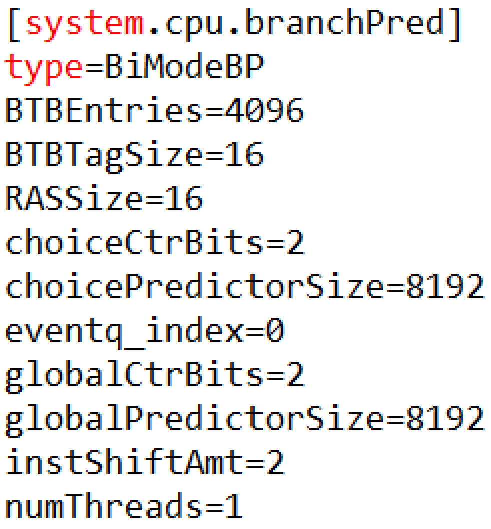
LocalBP	:	2bit Local Branch Predictor
BiModeBP	:	Global Behavioral Branch Predictor
Tournament	:	Combined/Hybrid Predictor

★ config.ini - Local



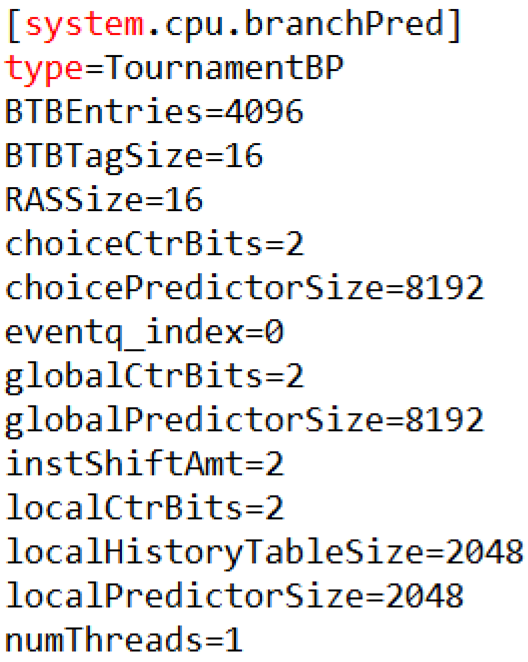
```
[system.cpu.branchPred]
type=LocalBP
BTBEntries=4096
BTBTagSize=16
RASSize=16
eventq_index=0
instShiftAmt=2
localCtrBits=2
localPredictorSize=2048
numThreads=1
```

★ config.ini - Bimode



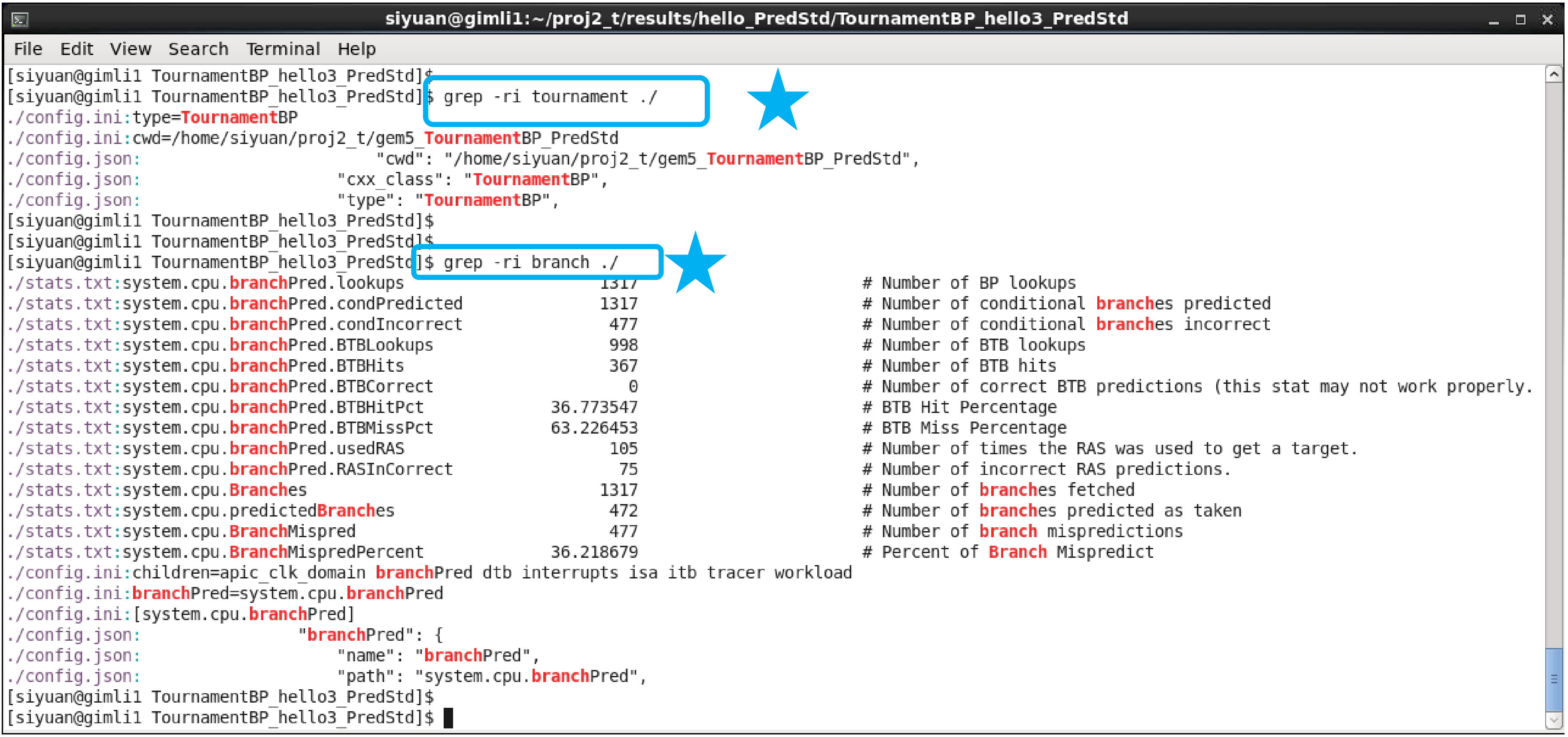
```
[system.cpu.branchPred]
type=BiModeBP
BTBEntries=4096
BTBTagSize=16
RASSize=16
choiceCtrBits=2
choicePredictorSize=8192
eventq_index=0
globalCtrBits=2
globalPredictorSize=8192
instShiftAmt=2
numThreads=1
```

★ config.ini - Tournament



```
[system.cpu.branchPred]
type=TournamentBP
BTBEntries=4096
BTBTagSize=16
RASSize=16
choiceCtrBits=2
choicePredictorSize=8192
eventq_index=0
globalCtrBits=2
globalPredictorSize=8192
instShiftAmt=2
localCtrBits=2
localHistoryTableSize=2048
localPredictorSize=2048
numThreads=1
```

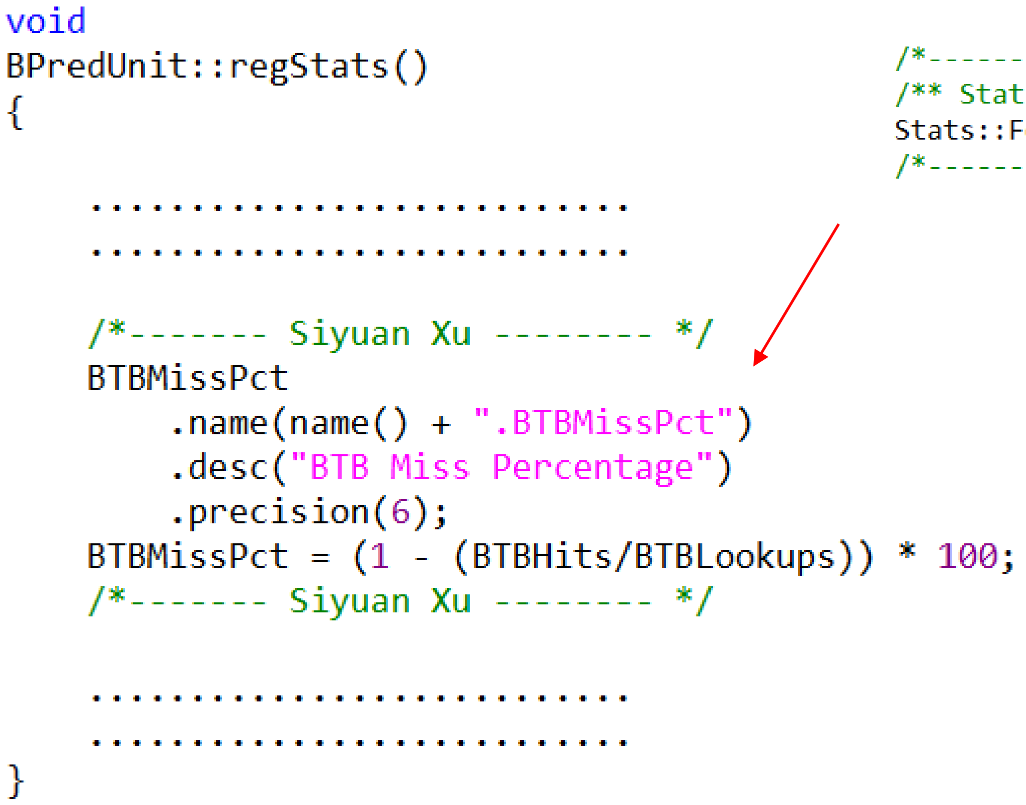
2. Result with **Tournament** for hello Bench


siyuan@gimli1:~/proj2_t/results/hello_PredStd/TournamentBP_hello3_PredStd
File Edit View Search Terminal Help

```
[siyuan@gimli1 TournamentBP_hello3_PredStd]$
[siyuan@gimli1 TournamentBP_hello3_PredStd]$ grep -ri tournament ./
./config.ini:type=TournamentBP
./config.ini:cwd=/home/siyuan/proj2_t/gem5_TournamentBP_PredStd
./config.json:                        "cwd": "/home/siyuan/proj2_t/gem5_TournamentBP_PredStd",
./config.json:                    "cxx_class": "TournamentBP",
./config.json:                    "type": "TournamentBP",
[siyuan@gimli1 TournamentBP_hello3_PredStd]$
[siyuan@gimli1 TournamentBP_hello3_PredStd]$
[siyuan@gimli1 TournamentBP_hello3_PredStd]$ grep -ri branch ./
./stats.txt:system.cpu.branchPred.lookups                    1317                       # Number of BP lookups
./stats.txt:system.cpu.branchPred.condPredicted              1317                       # Number of conditional branches predicted
./stats.txt:system.cpu.branchPred.condIncorrect               477                       # Number of conditional branches incorrect
./stats.txt:system.cpu.branchPred.BTBLookups                  998                       # Number of BTB lookups
./stats.txt:system.cpu.branchPred.BTBHits                     367                       # Number of BTB hits
./stats.txt:system.cpu.branchPred.BTBCorrect                    0                       # Number of correct BTB predictions (this stat may not work properly.
./stats.txt:system.cpu.branchPred.BTBHitPct             36.773547                       # BTB Hit Percentage
./stats.txt:system.cpu.branchPred.BTBMissPct            63.226453                       # BTB Miss Percentage
./stats.txt:system.cpu.branchPred.usedRAS                     105                       # Number of times the RAS was used to get a target.
./stats.txt:system.cpu.branchPred.RASInCorrect                 75                       # Number of incorrect RAS predictions.
./stats.txt:system.cpu.Branches                              1317                       # Number of branches fetched
./stats.txt:system.cpu.predictedBranches                      472                       # Number of branches predicted as taken
./stats.txt:system.cpu.BranchMispred                          477                       # Number of branch mispredictions
./stats.txt:system.cpu.BranchMispredPercent             36.218679                       # Percent of Branch Mispredict
./config.ini:children=apic_clk_domain branchPred dtb interrupts isa itb tracer workload
./config.ini:branchPred=system.cpu.branchPred
./config.ini:[system.cpu.branchPred]
./config.json:                "branchPred": {
./config.json:                    "name": "branchPred",
./config.json:                    "path": "system.cpu.branchPred",
[siyuan@gimli1 TournamentBP_hello3_PredStd]$
[siyuan@gimli1 TournamentBP_hello3_PredStd]$ 
```

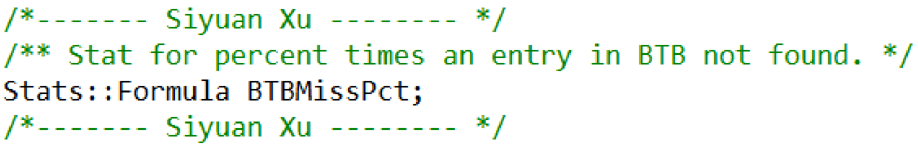
3.Adding BTB Miss Percentage

★ **$Gem5/src/cpu/pred/bpred_unit.cc**



```
void
BPredUnit::regStats()
{

    ...........................
    ...........................

    /*------- Siyuan Xu -------- */
    BTBMissPct
        .name(name() + ".BTBMissPct")
        .desc("BTB Miss Percentage")
        .precision(6);
    BTBMissPct = (1 - (BTBHits/BTBLookups)) * 100;
    /*------- Siyuan Xu -------- */

    ...........................
    ...........................
}
```

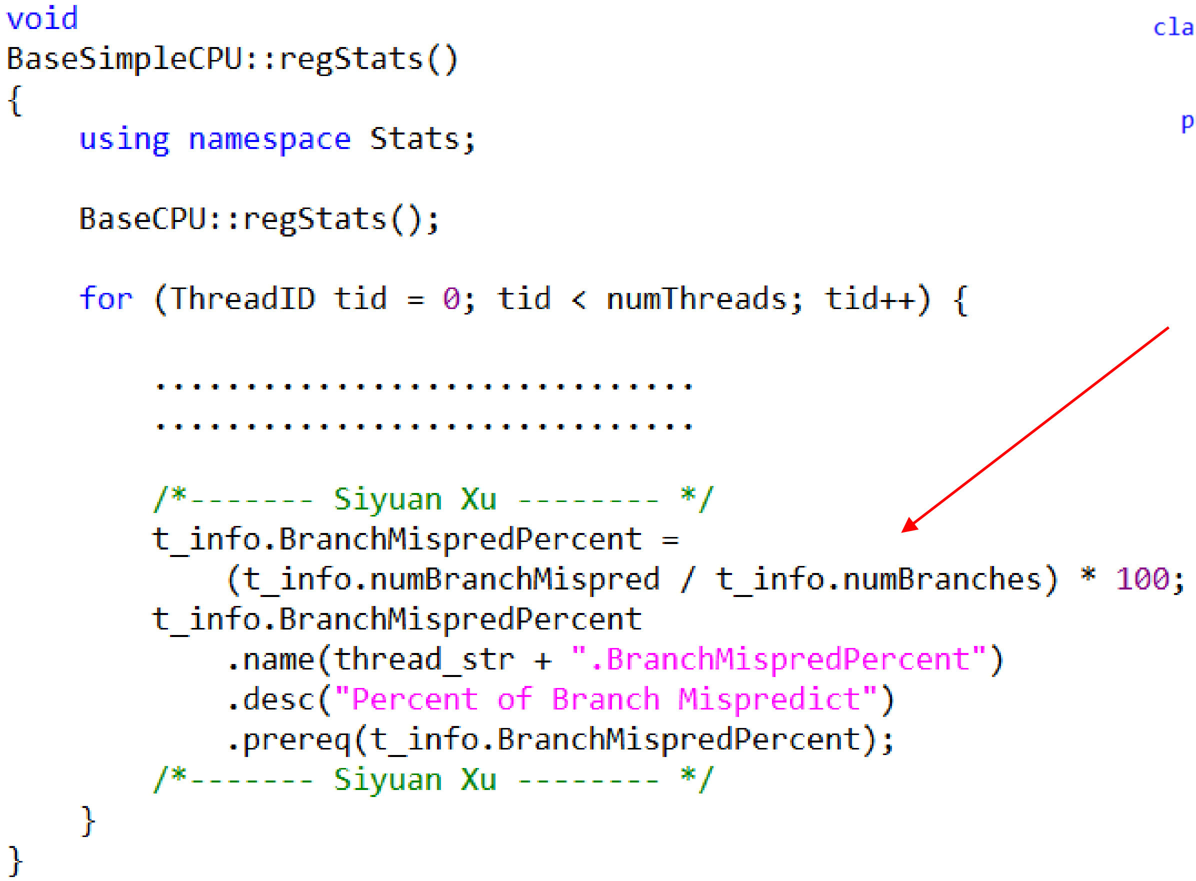
★ **$Gem5/src/cpu/pred/bpred_unit.hh**



```
/*------- Siyuan Xu -------- */
/** Stat for percent times an entry in BTB not found. */
Stats::Formula BTBMissPct;
/*------- Siyuan Xu -------- */
```

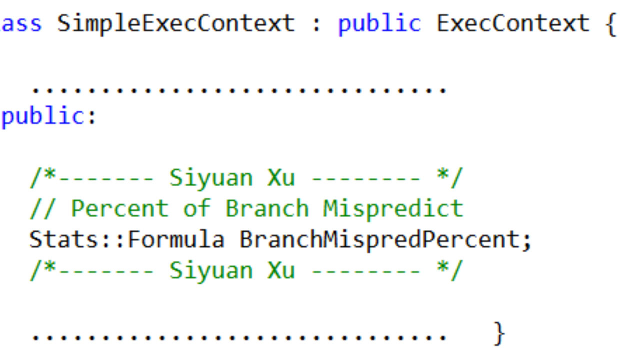
3.Adding extra resulting Parameters

★ **$Gem5/src/cpu/simple/base.cc**



```
void
BaseSimpleCPU::regStats()
{
    using namespace Stats;

    BaseCPU::regStats();

    for (ThreadID tid = 0; tid < numThreads; tid++) {

        ..............................
        ..............................

        /*------- Siyuan Xu -------- */
        t_info.BranchMispredPercent =
            (t_info.numBranchMispred / t_info.numBranches) * 100;
        t_info.BranchMispredPercent
            .name(thread_str + ".BranchMispredPercent")
            .desc("Percent of Branch Mispredict")
            .prereq(t_info.BranchMispredPercent);
        /*------- Siyuan Xu -------- */
    }
}
```

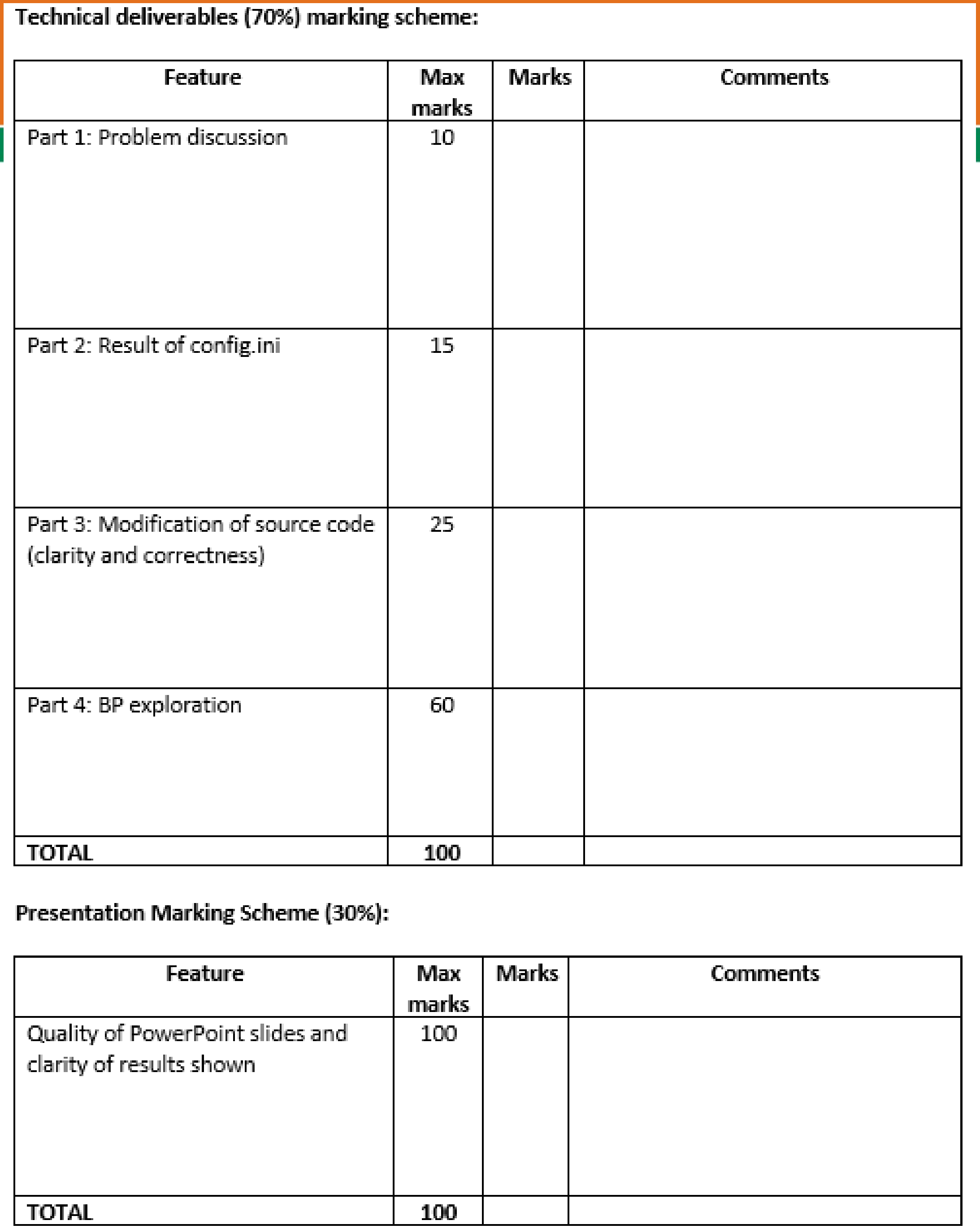
★ **$Gem5/src/cpu/simple/exec_context.hh**



```
class SimpleExecContext : public ExecContext {

    ..............................
  public:

    /*------- Siyuan Xu -------- */
    // Percent of Branch Mispredict
    Stats::Formula BranchMispredPercent;
    /*------- Siyuan Xu -------- */

    ..............................   }
```

4. Branch Prediction Exploration

- BTBEntries : 4096 -> 2048
- LocalBP localPredictorSize : 2048 -> 1024
- BiModeBP globalPredictorSize : 8192 -> 2048
- BiModeBP choicePredictorSize : 8192 -> 2048
- TournamentBP localPredictorSize : 2048 -> 1024
- TournamentBP globalPredictorSize: 8192 -> 4096
- TournamentBP choicePredictorSize: 8192 -> 4096



Technical deliverables (70%) marking scheme:

Feature	Max marks	Marks	Comments
Part 1: Problem discussion	10		
Part 2: Result of config.ini	15		
Part 3: Modification of source code (clarity and correctness)	25		
Part 4: BP exploration	60		
TOTAL	**100**		

Presentation Marking Scheme (30%):

Feature	Max marks	Marks	Comments
Quality of PowerPoint slides and clarity of results shown	100		
TOTAL	**100**		