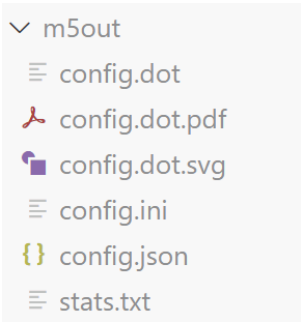
Assingment 3
Nama : Kosmas Rio Legowo
NIM : 23/512012/PA/21863

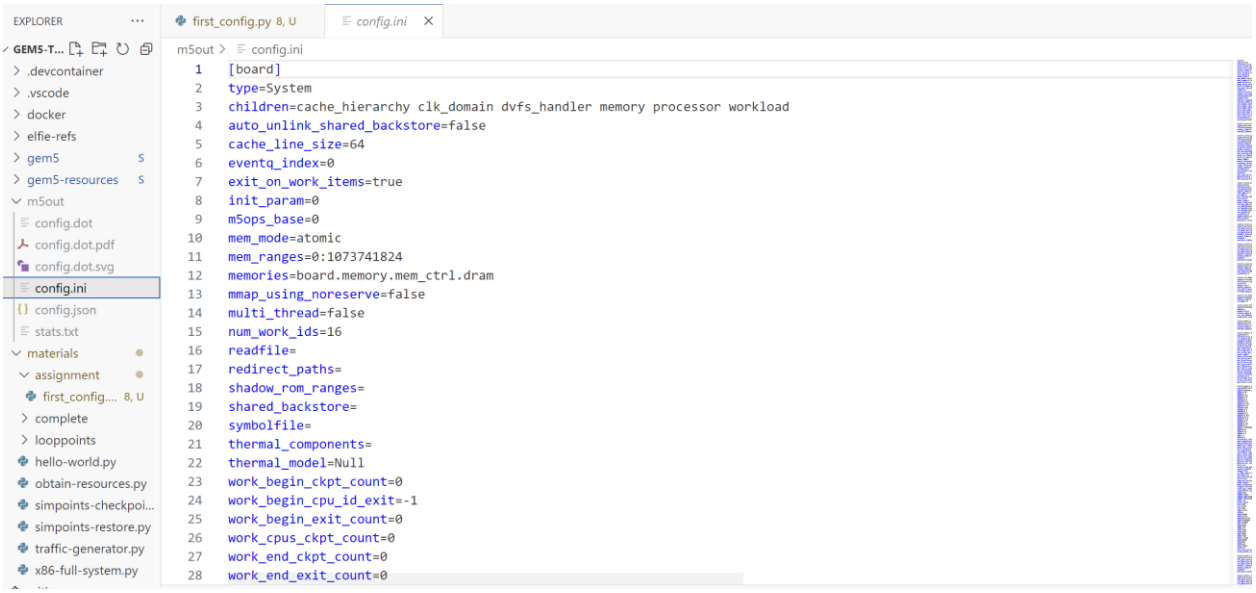
2.6 First Experiment

```
root@codespaces-1c2f57:/workspaces/gem5-tutorial-codespace# gem5 materials/assignment/first_config.py
gem5 Simulator System.  https://www.gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 version [DEVELOP-FOR-23.0]
gem5 compiled Feb 25 2023 19:25:04
gem5 started Sep 11 2024 02:02:56
gem5 executing on codespaces-1c2f57, pid 2662
command line: gem5 materials/assignment/first_config.py

Resource 'x86-hello64-static' was not found locally. Downloading to '/root/.cache/gem5/x86-hello64-static'...
Finished downloading resource 'x86-hello64-static'.
warn: The `CustomResource` class is deprecated. Please use an `AbstractResource` subclass instead.
warn: The simulate package is still in a beta state. The gem5 project does not guarantee the APIs within this package will remain consistent across upcoming r
eleases.
Global frequency set at 1000000000000 ticks per second
build/ALL/mem/dram_interface.cc:690: warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (1024 Mbytes)
build/ALL/base/statistics.hh:279: warn: One of the stats is a legacy stat. Legacy stat is a stat that does not belong to any statistics::Group. Legacy stat is
 deprecated.
0: board.remote_gdb: listening for remote gdb on port 7000
build/ALL/sim/simulate.cc:194: info: Entering event queue @ 0.  Starting simulation...
build/ALL/sim/syscall_emul.hh:1014: warn: readlink() called on '/proc/self/exe' may yield unexpected results in various settings.
      Returning '/root/.cache/gem5/x86-hello64-static'
build/ALL/sim/mem_state.cc:443: info: Increasing stack size by one page.
Hello world!
```

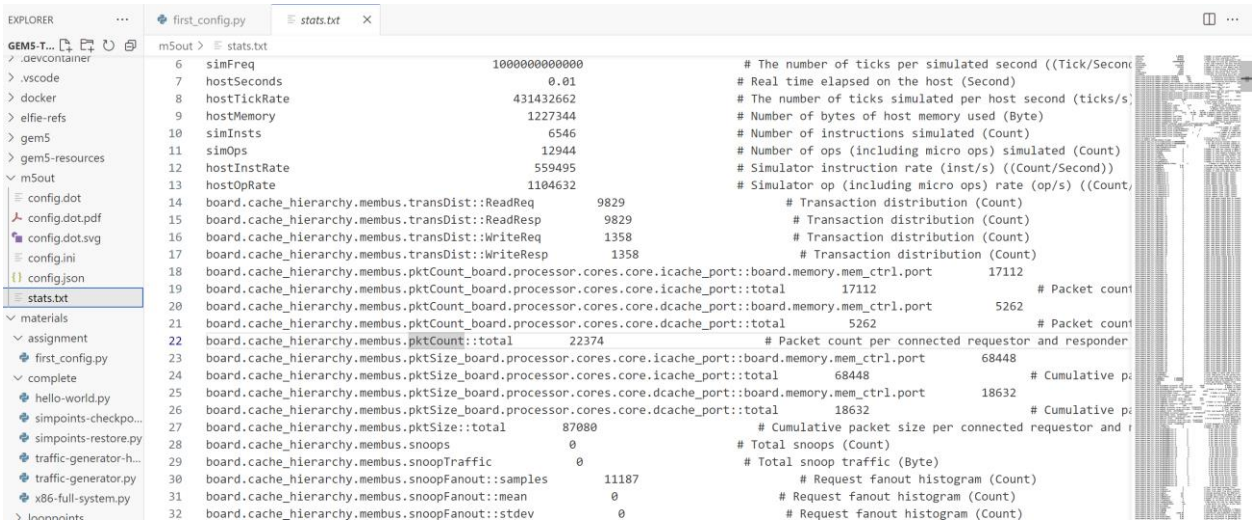
2.7 Understanding gem5 Statistics and Output



2.7.1 config.ini



2.7.2 stats.txt



2.8 Second Experiment

Output

```
root@codespaces-1c2f57:/workspaces/gem5-tutorial-codespace# gem5 materials/assignment/second_config.py
gem5 Simulator System.  https://www.gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 version [DEVELOP-FOR-23.0]
gem5 compiled Feb 25 2023 19:25:04
gem5 started Sep 11 2024 02:41:12
gem5 executing on codespaces-1c2f57, pid 7193
command line: gem5 materials/assignment/second_config.py

Global frequency set at 1000000000000 ticks per second
build/ALL/mem/dram_interface.cc:690: warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (512 Mbytes)
build/ALL/base/statistics.hh:279: warn: One of the stats is a legacy stat. Legacy stat is a stat that does not belong to any statistics::Group. Legacy stat is
 deprecated.
0: system.remote_gdb: listening for remote gdb on port 7000
 Beginning simulation !
build/ALL/sim/simulate.cc:194: info: Entering event queue @ 0.  Starting simulation...
Hello world!
 Exiting @tick 462979000 because exiting with last active thread context
root@codespaces-1c2f57:/workspaces/gem5-tutorial-codespace#
```

2.9 Third Experiment

Source code for caches

caches.py

```
import m5
from m5.objects import Cache

class L1Cache(Cache):
    """Simple L1 Cache with default values"""
#
    assoc = 2
    tag_latency = 2
    data_latency = 2
    response_latency = 2
    mshrs = 4
    tgts_per_mshr = 20

    def __init__(self, options=None):
        super().__init__()
        pass

    def connectBus(self, bus):
        """Connect this cache to a memory-side bus"""
        self.mem_side = bus.cpu_side_ports

    def connectCPU(self, cpu):
        """Connect this cache's port to a CPU-side port
        This must be defined in a subclass"""
        raise NotImplementedError


class L1ICache(L1Cache):
    """Simple L1 instruction cache with default values"""

    # Set the default size
    size = "16kB"

    def __init__(self, opts=None):
        super().__init__(opts)
        if not opts or not opts['l1i_size']:
            return
        self.size = opts['l1i_size']

    def connectCPU(self, cpu):
        """Connect this cache's port to a CPU icache port"""
        self.cpu_side = cpu.icache_port
```

```
class L1DCache(L1Cache):
    """Simple L1 data cache with default values"""

    # Set the default size
    size = "64kB"

    def __init__(self, opts=None):
        super().__init__(opts)
        if not opts or not opts['l1d_size']:
            return
        self.size = opts['l1d_size']

    def connectCPU(self, cpu):
        """Connect this cache's port to a CPU dcache port"""
        self.cpu_side = cpu.dcache_port


class L2Cache(Cache):
    """Simple L2 Cache with default values"""

    # Default parameters
    size = "256kB"
    assoc = 8
    tag_latency = 20
    data_latency = 20
    response_latency = 20
    mshrs = 20
    tgts_per_mshr = 12

    def __init__(self, opts=None):
        super().__init__()
        if not opts or not opts['l2_size']:
            return
        self.size = opts['l2_size']

    def connectCPUSideBus(self, bus):
        self.cpu_side = bus.mem_side_ports

    def connectMemSideBus(self, bus):
        self.mem_side = bus.cpu_side_ports
```

Source code for the updated configuration

third_config.py

```
import m5
from m5.objects import *
from caches import *

system = System()

system.clk_domain = SrcClockDomain()
system.clk_domain.clock = '1GHz'
system.clk_domain.voltage_domain = VoltageDomain()

system.mem_mode = 'timing'
system.mem_ranges = [AddrRange('512 MB')]

system.cpu = X86TimingSimpleCPU()

args = {
    'l1i_size': '16 kB',
    'l1d_size': '64 kB',
    'l2_size': '256 kB'
}

# Create an L1 instruction and data cache
system.cpu.icache = L1ICache(args)
system.cpu.dcache = L1DCache(args)

# Connect the instruction and data caches to the CPU
system.cpu.icache.connectCPU(system.cpu)
system.cpu.dcache.connectCPU(system.cpu)

# Create a memory bus, a coherent crossbar, in this case
system.l2bus = L2XBar()

# Hook the CPU ports up to the l2bus
system.cpu.icache.connectBus(system.l2bus)
system.cpu.dcache.connectBus(system.l2bus)

# Create an L2 cache and connect it to the l2bus
system.l2cache = L2Cache(args)
system.l2cache.connectCPUSideBus(system.l2bus)

# Create a memory bus
system.membus = SystemXBar()
```

```
# Connect the L2 cache to the membus
system.l2cache.connectMemSideBus(system.membus)

# create the interrupt controller for the CPU
system.cpu.createInterruptController()
system.cpu.interrupts[0].pio = system.membus.mem_side_ports
system.cpu.interrupts[0].int_requestor = system.membus.cpu_side_ports
system.cpu.interrupts[0].int_responder = system.membus.mem_side_ports

# Connect the system up to the membus
system.system_port = system.membus.cpu_side_ports

# Create a DDR3 memory controller
system.mem_ctrl = MemCtrl()
system.mem_ctrl.dram = DDR3_1600_8x8()
system.mem_ctrl.dram.range = system.mem_ranges[0]
system.mem_ctrl.port = system.membus.mem_side_ports

binary = "gem5/tests/test-progs/hello/bin/x86/linux/hello"
system.workload = SEWorkload.init_compatible(binary)

# Create a process for a simple " Hello World "application
process = Process ()
# Set the command
# cmd is a list which begins with the executable ( likeargv )
process . cmd = [ binary ]
# Set the cpu to use the process as its workload and create thread contexts
system . cpu . workload = process
system . cpu . createThreads ()

# set up the root SimObject and start the simulation
root = Root ( full_system = False , system = system )
# instantiate all of the objects we have created above
m5 . instantiate ()

print (" Beginning simulation !")
exit_event = m5 . simulate ()
print (" Exiting @tick %i because %s" % ( m5 . curTick () ,
    exit_event . getCause () )
```

Outputs:

2.10 Gem5 in Computer Architecture Research

1. D. Ojha and S. Dwarkadas, "TimeCache: Using Time to Eliminate Cache Side Channels when Sharing Software," 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), Valencia, Spain, 2021, pp. 375-387, doi: 10.1109/ISCA52012.2021.00037.
Summary:
Paper ini mempunyai tujuan untuk mengurangi serangan side-channel cache di sistem multi-tenant dengan memperkenalkan TimeCache, yang melacak penggunaan cache line menggunakan timestamp. Metode ini diuji menggunakan simulasi gem5 dan hardware sungguhan yang kemudian berhasil mencegah serangan seperti flush+reload tanpa menimbulkan kerentanan baru. TimeCache juga menunjukkan overhead kinerja minimal (sekitar 1,13%) sambil memberikan keamanan yang kuat.

2. L. Moody *et al.*, "Speculative Code Compaction: Eliminating Dead Code via Speculative Microcode Transformations," *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Chicago, IL, USA, 2022, pp. 162-180, doi: 10.1109/MICRO56248.2022.00024.
Summary: Paper ini berfokus pada optimalisasi mikrocode dengan menghilangkan dead code melalui transformasi spekulatif. Dilakukan simulasi menggunakan gem5 untuk mengevaluasi performa dari Speculative Code Compaction. Simulasi ini menunjukkan bahwa pendekatan ini memberikan penghematan energi yang signifikan tanpa mengorbankan kecepatan eksekusi.