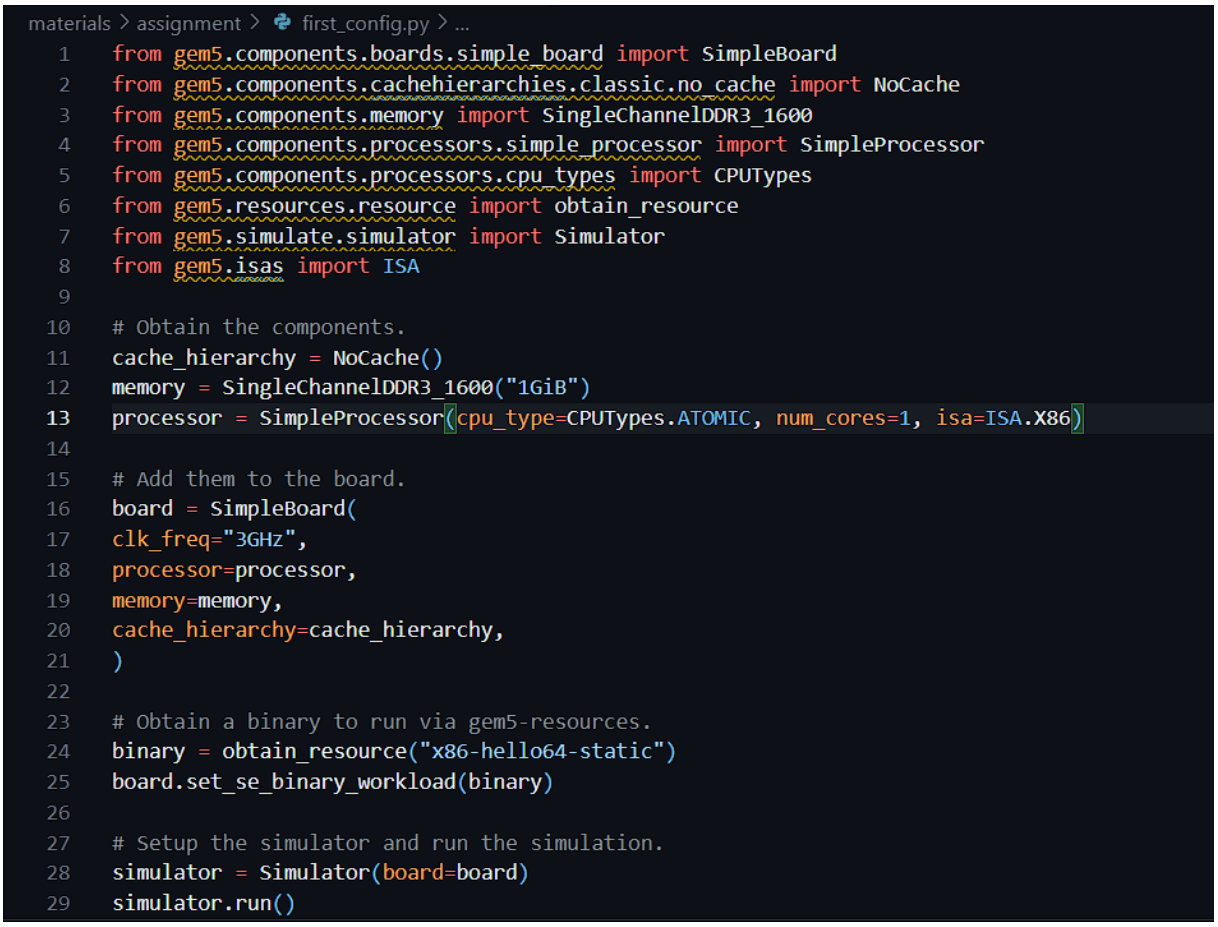
Nama: Daffa Harikhsan
NIM: 23/513044/PA/21918

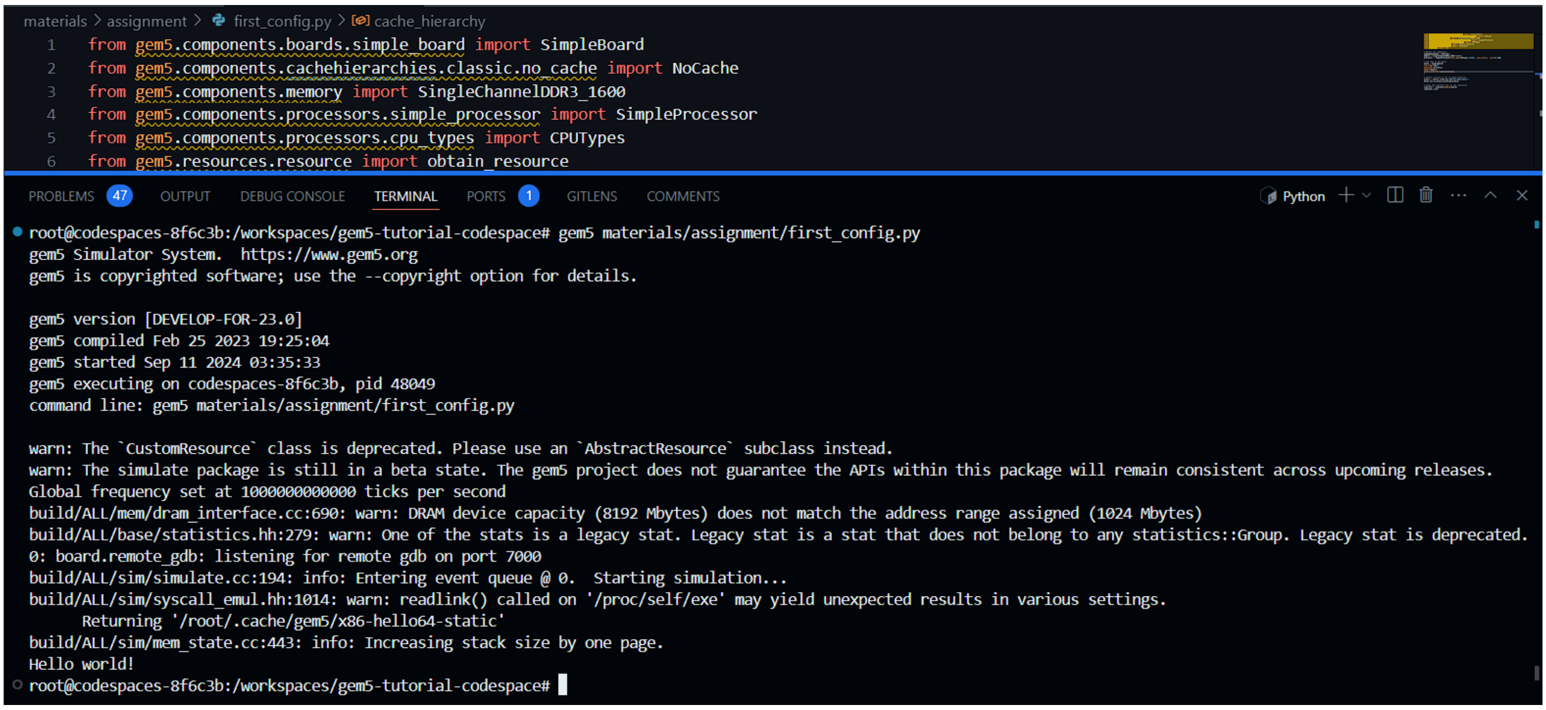
Tugas 3

1. Task 1 (First_config)



materials > assignment > first_config.py > ...

```python
from gem5.components.boards.simple_board import SimpleBoard
from gem5.components.cachehierarchies.classic.no_cache import NoCache
from gem5.components.memory import SingleChannelDDR3_1600
from gem5.components.processors.simple_processor import SimpleProcessor
from gem5.components.processors.cpu_types import CPUTypes
from gem5.resources.resource import obtain_resource
from gem5.simulate.simulator import Simulator
from gem5.isas import ISA

# Obtain the components.
cache_hierarchy = NoCache()
memory = SingleChannelDDR3_1600("1GiB")
processor = SimpleProcessor(cpu_type=CPUTypes.ATOMIC, num_cores=1, isa=ISA.X86)

# Add them to the board.
board = SimpleBoard(
clk_freq="3GHz",
processor=processor,
memory=memory,
cache_hierarchy=cache_hierarchy,
)

# Obtain a binary to run via gem5-resources.
binary = obtain_resource("x86-hello64-static")
board.set_se_binary_workload(binary)

# Setup the simulator and run the simulation.
simulator = Simulator(board=board)
simulator.run()
```



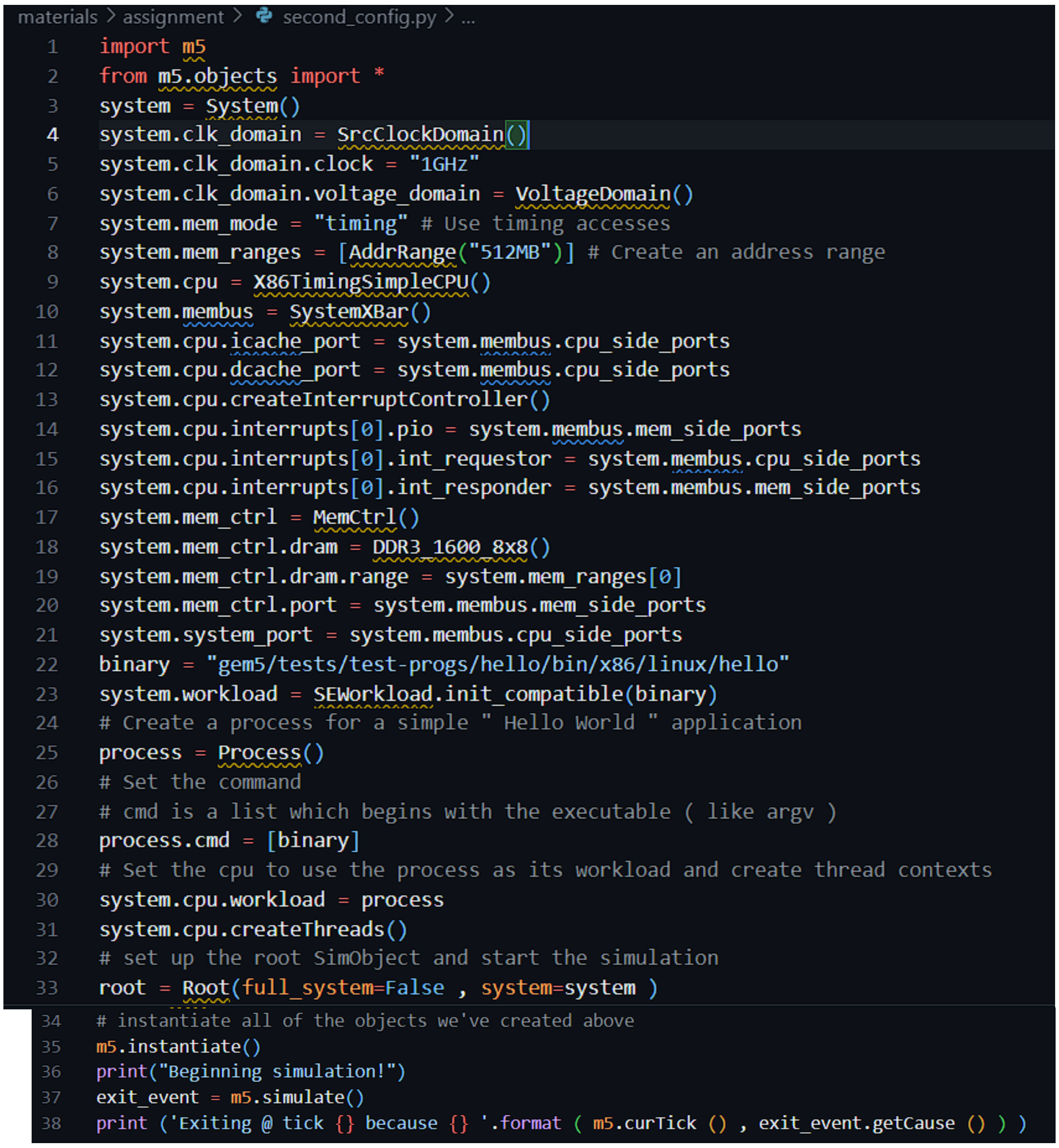
materials > assignment > first_config.py > cache_hierarchy

```python
from gem5.components.boards.simple_board import SimpleBoard
from gem5.components.cachehierarchies.classic.no_cache import NoCache
from gem5.components.memory import SingleChannelDDR3_1600
from gem5.components.processors.simple_processor import SimpleProcessor
from gem5.components.processors.cpu_types import CPUTypes
from gem5.resources.resource import obtain_resource
```

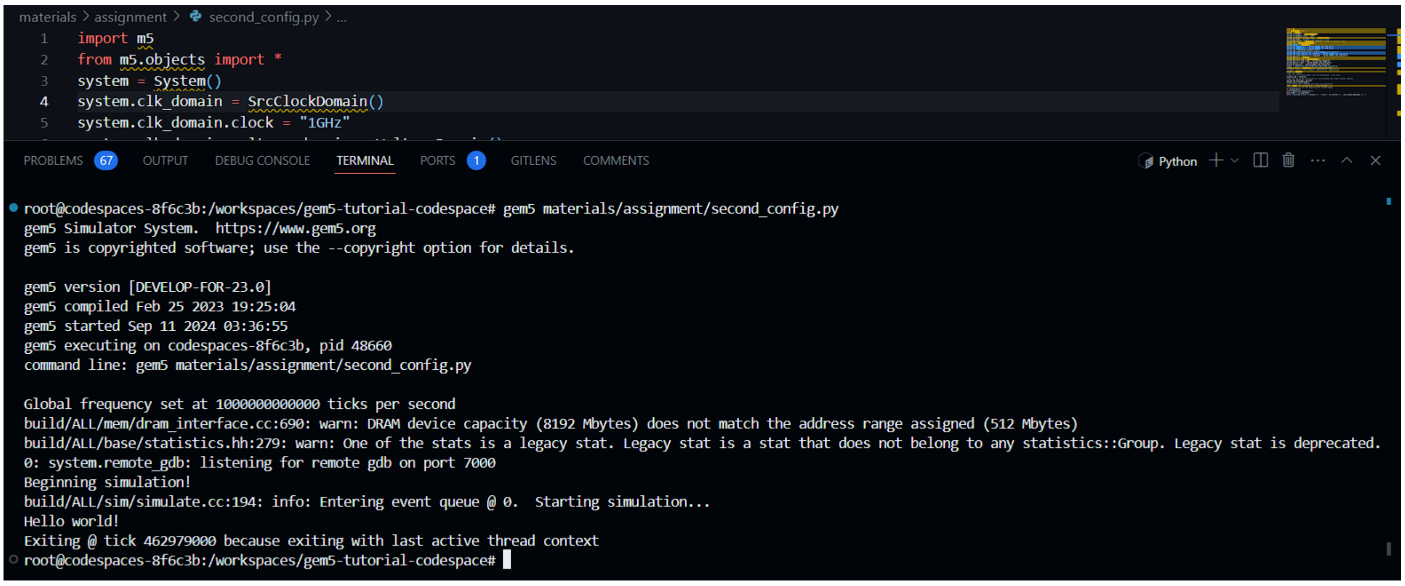
```
root@codespaces-8f6c3b:/workspaces/gem5-tutorial-codespace# gem5 materials/assignment/first_config.py
gem5 Simulator System.  https://www.gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 version [DEVELOP-FOR-23.0]
gem5 compiled Feb 25 2023 19:25:04
gem5 started Sep 11 2024 03:35:33
gem5 executing on codespaces-8f6c3b, pid 48049
command line: gem5 materials/assignment/first_config.py

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 releases.
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!
root@codespaces-8f6c3b:/workspaces/gem5-tutorial-codespace#
```

Penjelasan:
Kode tersebut menjelaskan bagaimana output “Hello World” tercipta, ini menjalankan simulasi sistem komputer menggunakan gem5. Tetapi ditunjukkan peringatan beberapa terkait konfigurasi, seperti kapasitas memuri yang tidak sesuai dengan pengaturan, penggunaan kelas yang sudah usang, dll. Simulasi ini juga mensimulasikan sebuah prosesor x86 satu core, dengan memori DDR3 1 GiB, dan tanpa cache.

2. Task 2 (Second_config)



```python
import m5
from m5.objects import *
system = System()
system.clk_domain = SrcClockDomain()
system.clk_domain.clock = "1GHz"
system.clk_domain.voltage_domain = VoltageDomain()
system.mem_mode = "timing" # Use timing accesses
system.mem_ranges = [AddrRange("512MB")] # Create an address range
system.cpu = X86TimingSimpleCPU()
system.membus = SystemXBar()
system.cpu.icache_port = system.membus.cpu_side_ports
system.cpu.dcache_port = system.membus.cpu_side_ports
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
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
system.system_port = system.membus.cpu_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 ( like argv )
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've created above
m5.instantiate()
print("Beginning simulation!")
exit_event = m5.simulate()
print ('Exiting @ tick {} because {} '.format ( m5.curTick () , exit_event.getCause () ) )
```

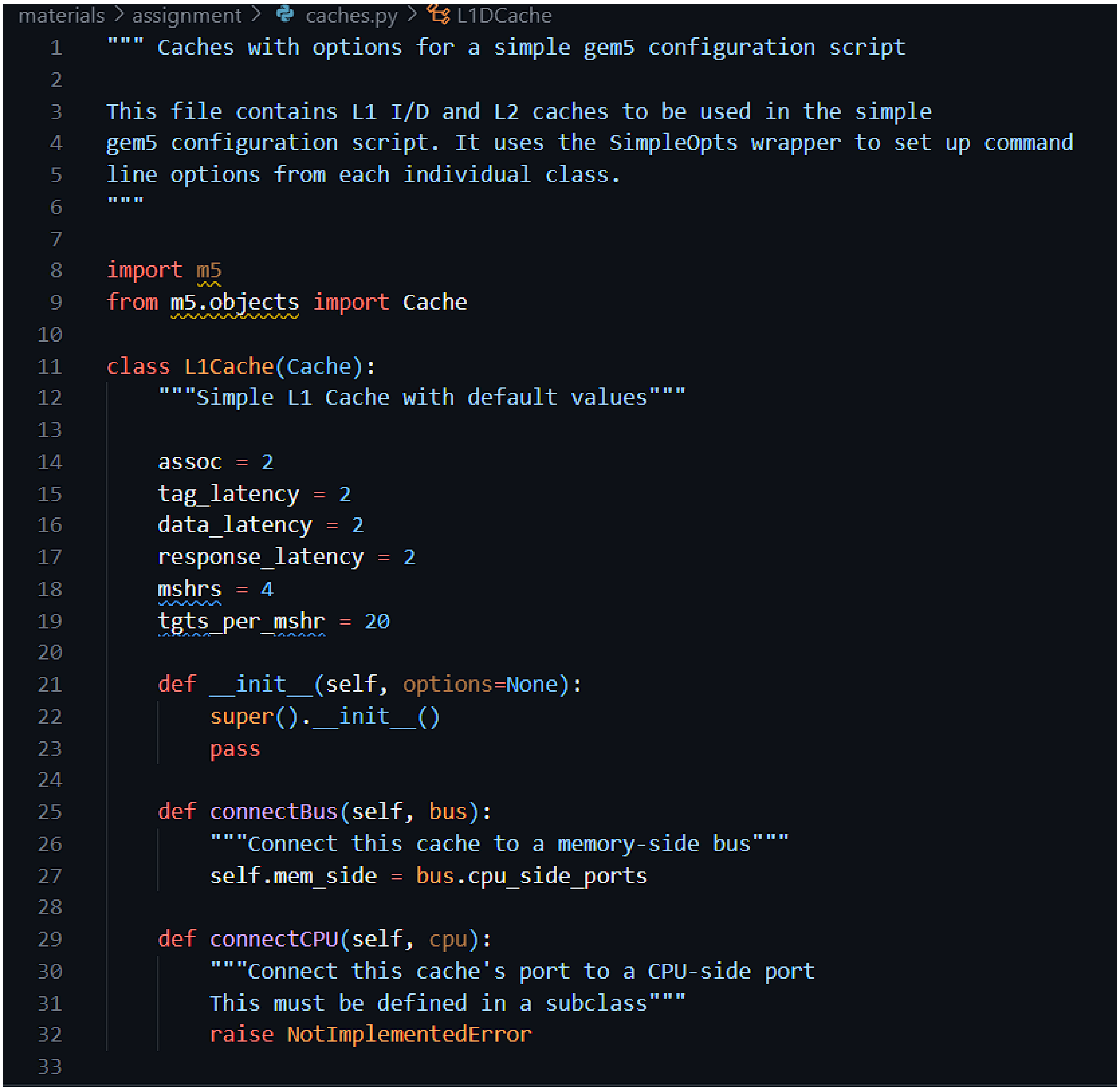


```python
import m5
from m5.objects import *
system = System()
system.clk_domain = SrcClockDomain()
system.clk_domain.clock = "1GHz"
```

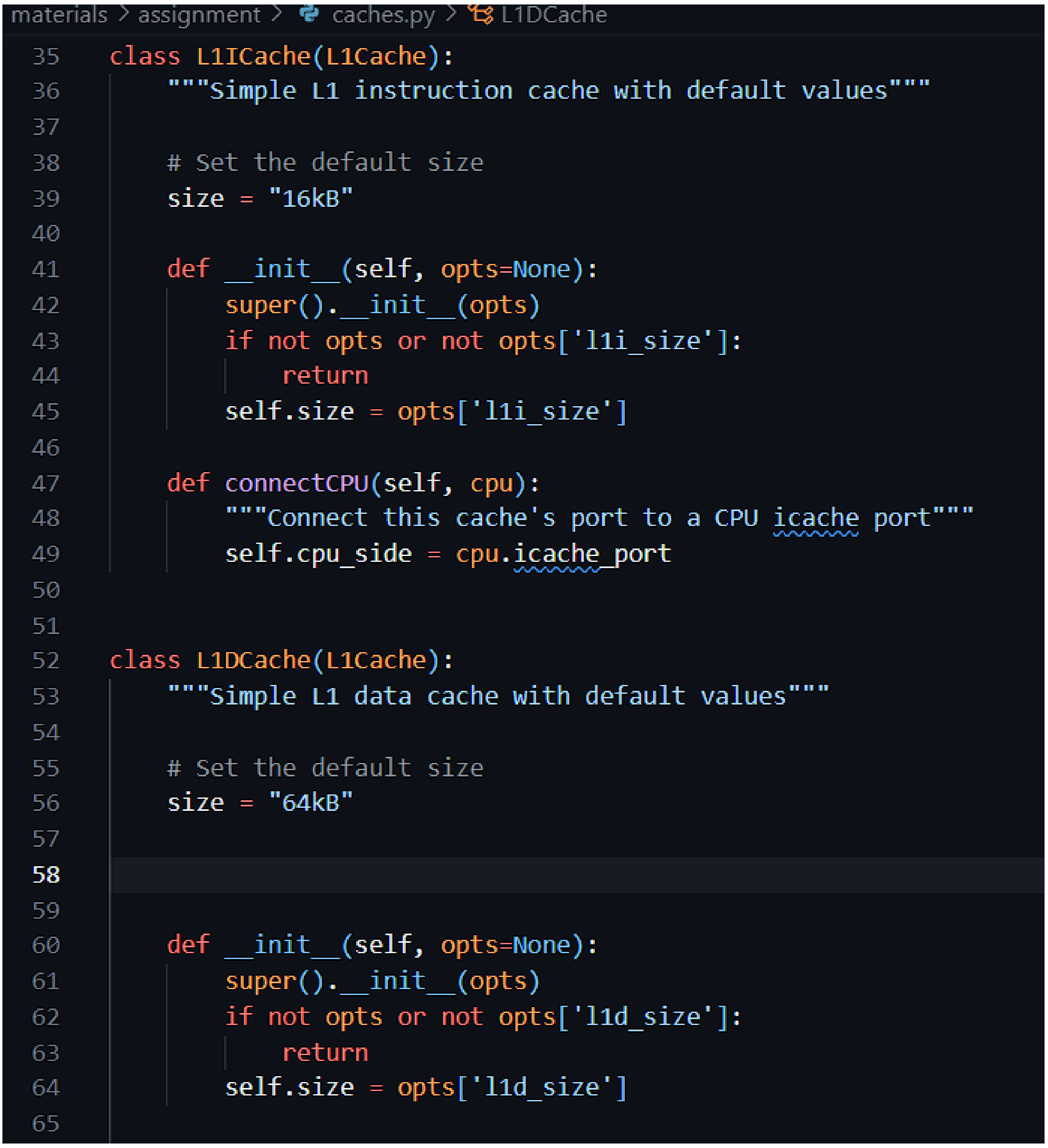
```
root@codespaces-8f6c3b:/workspaces/gem5-tutorial-codespace# gem5 materials/assignment/second_config.py
gem5 Simulator System.  https://www.gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 version [DEVELOP-FOR-23.0]
gem5 compiled Feb 25 2023 19:25:04
gem5 started Sep 11 2024 03:36:55
gem5 executing on codespaces-8f6c3b, pid 48660
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-8f6c3b:/workspaces/gem5-tutorial-codespace#
```

Penjelasan:
Kode tersebut menjelaskan bagaimana output “Hello World” Tercipta menggunakan simulasi gem5 dengan menggunakan model CPU X86TimingSimpleCPU, simulasi ini selesai saat program selesai dijalankan pada tick 462979000. Terdapat beberapa peringatan mengenai ketidaksesuaian antara kapasitas DRAM (8192 MB) dan alamat memori yang diperintahkan (512 MB).

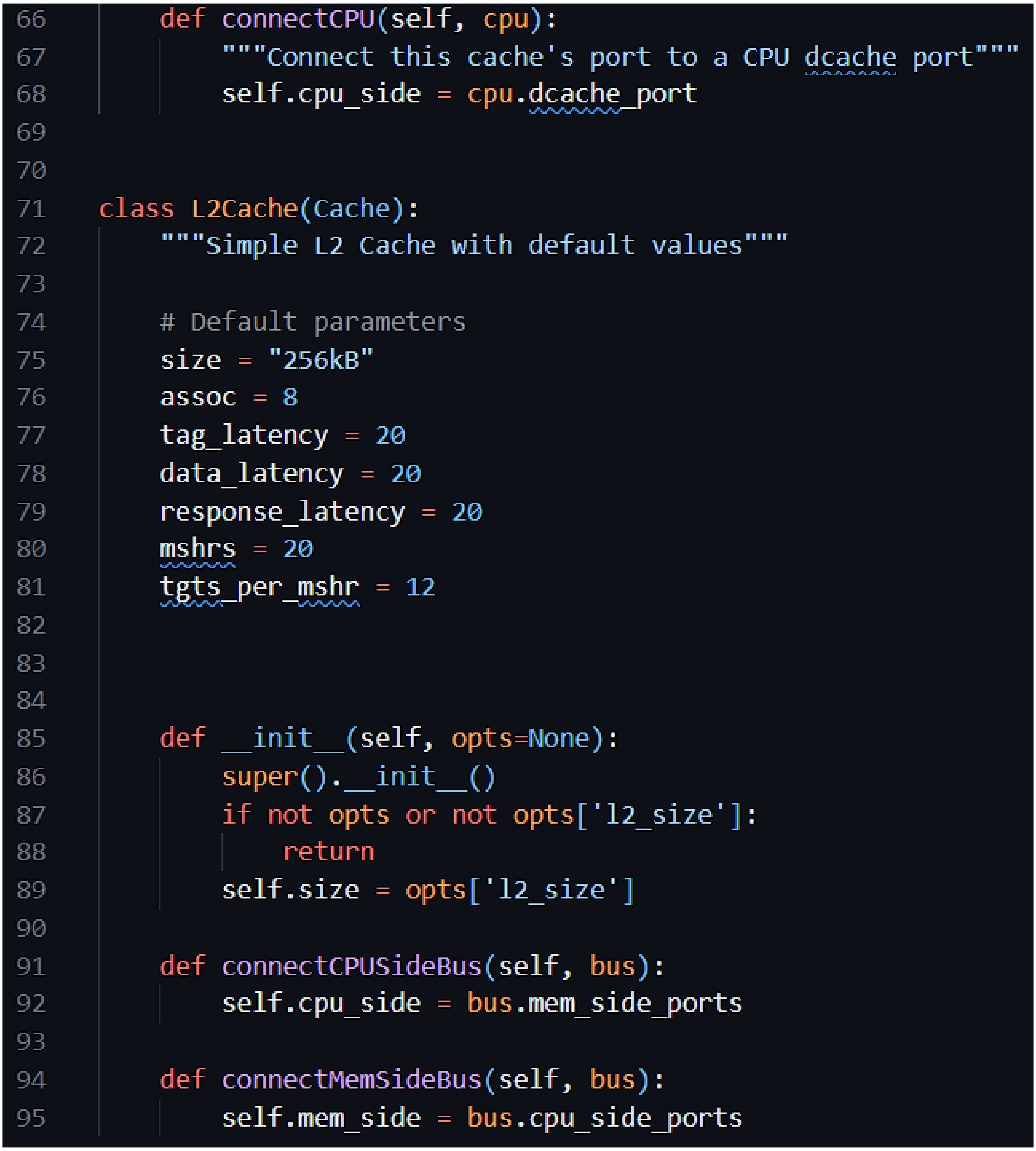
3. Task 3 (Third_config)



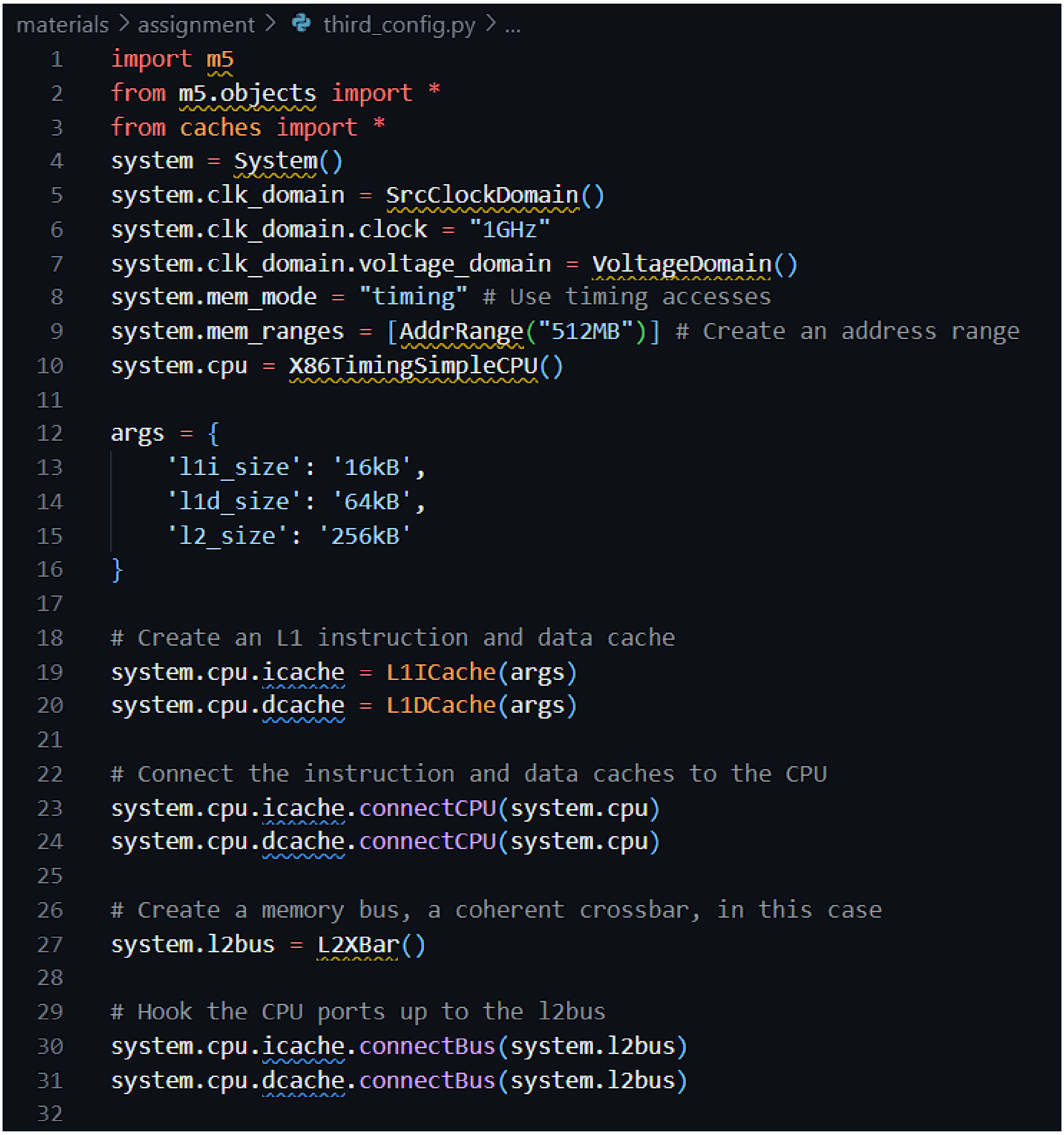
```python
""" Caches with options for a simple gem5 configuration script

This file contains L1 I/D and L2 caches to be used in the simple
gem5 configuration script. It uses the SimpleOpts wrapper to set up command
line options from each individual class.
"""

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

```


materials > assignment > caches.py > L1DCache

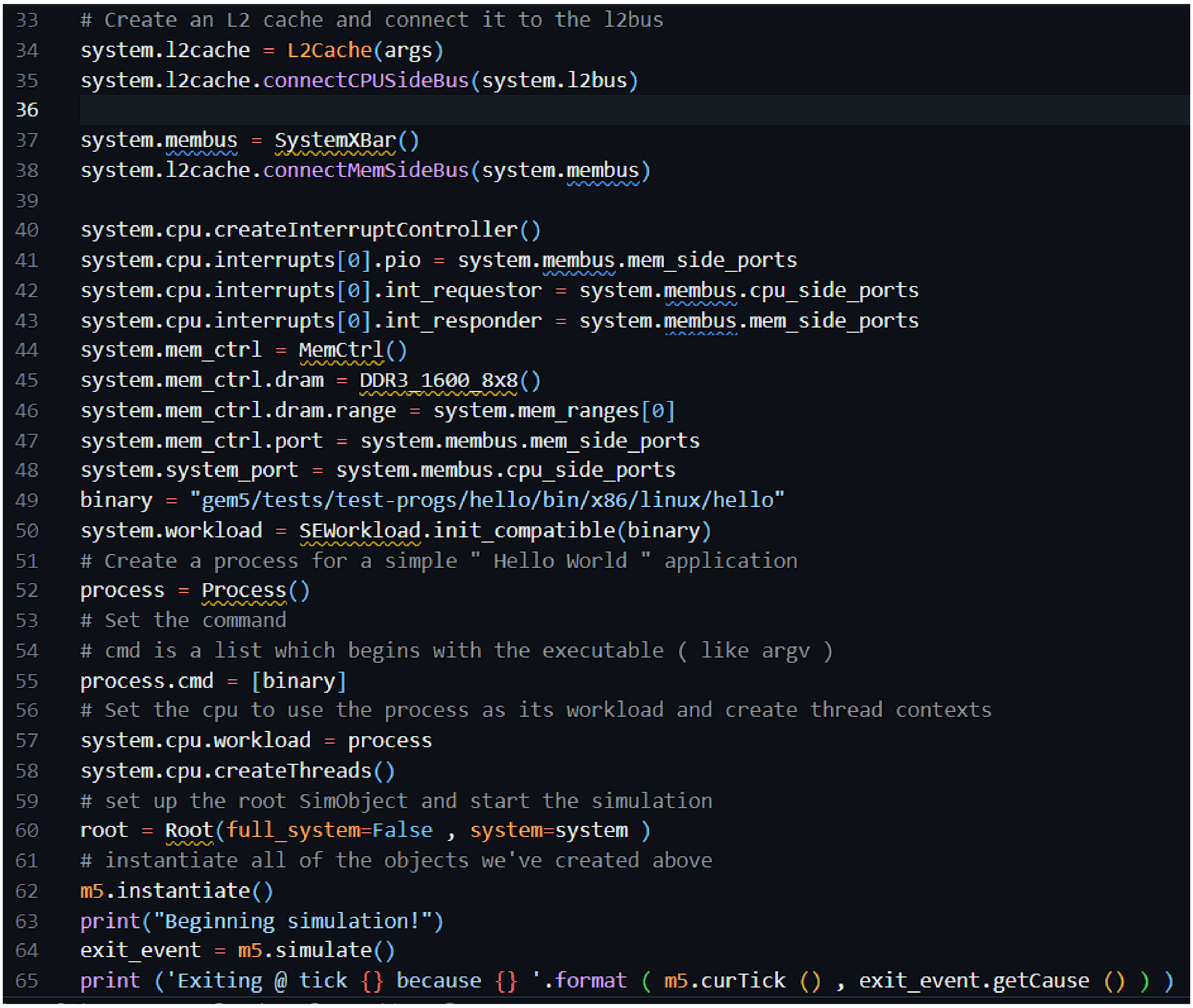
```python
class L1ICache(L1Cache):
    """Simple L1 instruction cache with default values"""

    # 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']

```



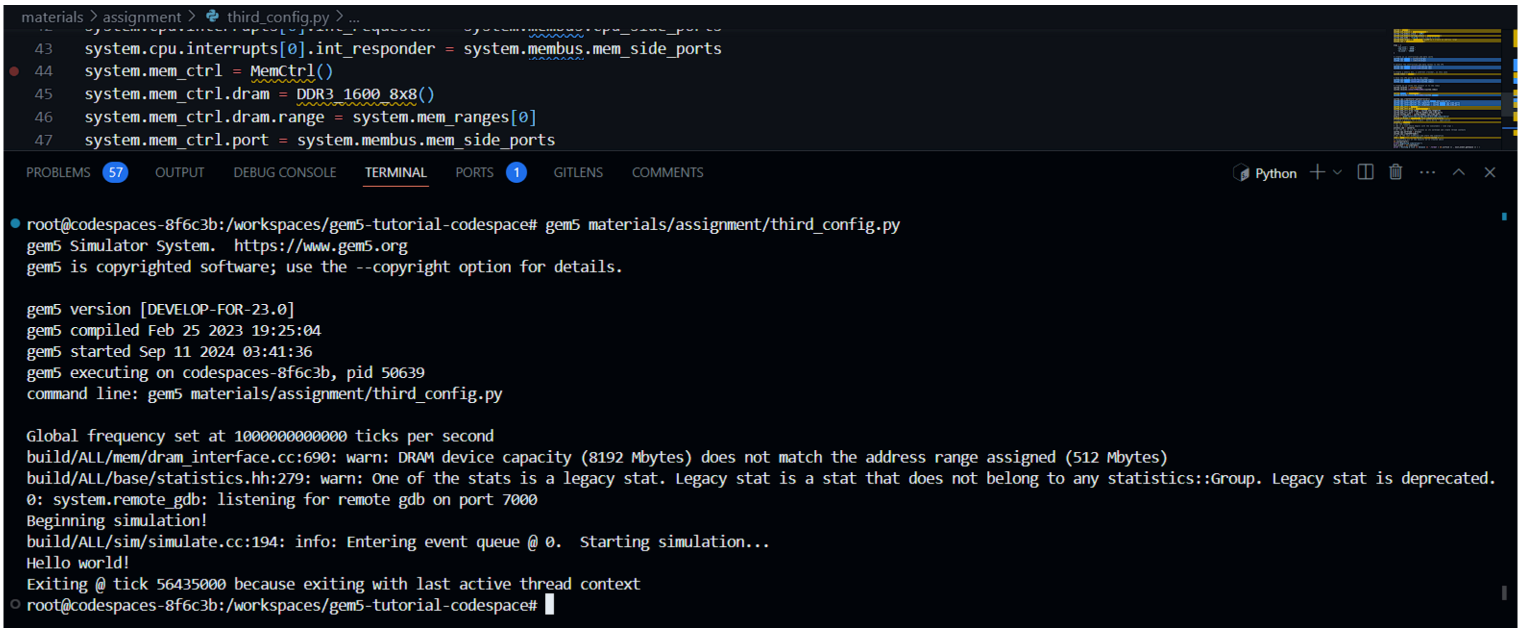
```python
    def connectCPU(self, cpu):
        """Connect this cache's port to a CPU dcache port"""
        self.cpu_side = cpu.dcache_port


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
```



```python
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" # Use timing accesses
system.mem_ranges = [AddrRange("512MB")] # Create an address range
system.cpu = X86TimingSimpleCPU()

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

# 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)

```



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

system.membus = SystemXBar()
system.l2cache.connectMemSideBus(system.membus)

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
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
system.system_port = system.membus.cpu_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 ( like argv )
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've created above
m5.instantiate()
print("Beginning simulation!")
exit_event = m5.simulate()
print ('Exiting @ tick {} because {} '.format ( m5.curTick () , exit_event.getCause () ) )
```



```python
system.cpu.interrupts[0].int_responder = system.membus.mem_side_ports
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
```

```
root@codespaces-8f6c3b:/workspaces/gem5-tutorial-codespace# gem5 materials/assignment/third_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 03:41:36
gem5 executing on codespaces-8f6c3b, pid 50639
command line: gem5 materials/assignment/third_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 56435000 because exiting with last active thread context
root@codespaces-8f6c3b:/workspaces/gem5-tutorial-codespace# 
```

Penjelasan:
Kode tersebut menjelaskan bagaimana output “Hello World” Tercipta menggunakan simulasi gem5, pada Task 3 sendiri juga merupakan implementasi untuk Cache, terdapat peringatan tentang ketidakcocokan kapasitas memori DRAM. Simulasi dimulai dengan benar, menampilkan "Hello world" dan berhenti dengan normal setelah semua proses selesai, berhenti di tick 56435000 karena tidak ada lagi konteks thread aktif yang berjalan, artinya semua proses telah selesai.

Paper:

Micro:
https://www.samxi.org/papers/shao_micro2016.pdf

Rangkuman:
Paper tersebut membahas tentang pentingnya co-design antara mikroarsitektur akselerator dan parameter platform SoC, untuk mencapai desain akselerator yang lebih efisien dan seimbang. Para peneliti disini memperkenalkan Gem5-Aladdin yang berupa simulator SoCyang menangkap interaksi dinamis antara akselerator dan platform SoC, memungkinkan eksplorasi co-design akselerator. Dengan adanya Gem5-Aladdin memungkinkan optimasi desain yang lebih baik, meningkatkan efisiensi akselerator hinggal 74% dibandingkan dengan optimasi akselerator secara terpisah.

ISCA:
https://www.pdl.cmu.edu/PDL-FTP/associated/perspective_isca24.pdf

Rangkuman:
Paper tersebut membahas mengenai “Perspective”, yakni sebuah kerangka kerja yang menggunakan DSV (Data Speculation Views dan ISV (Instruction Speculation Views) untuk meningkatkan keamanan spekulasi di OS (Sistem Operasi) dari serangan aktif pasif. Temuan utama menunjukkan bahwa metode ini mampu mengurangi serangan hingga 95% dan hanya menimbulkan overhead performa sebesar 1,2% dalam aplikasi data center. Ini memungkinkan perlindungan lebih baik dari serangan dan mempercepat proses audit kernel, sambil mempertahankan performa yang hampir sama dengan overhead yang sangat minimal.