

[illegible]

```

186 def main():
187     parser = argparse.ArgumentParser(epilog=__doc__)
188
189     parser.add_argument("commands_to_run", metavar="command(s)", nargs="*",
190                         help="Command(s) to run")
191     parser.add_argument("--cpu", type=str, choices=cpu_types.keys(),
192                         default="atomic",
193                         help="CPU model to use")
194     parser.add_argument("--cpu-freq", type=str, default="4GHz")
195     parser.add_argument("--num-cores", type=int, default=1,
196                         help="Number of CPU cores")
197     parser.add_argument("--mem-type", default="DDR3_1600_8x8",
198                         choices=ObjectList.mem_list.get_names(),
199                         help="type of memory to use")
200     parser.add_argument("--mem-channels", type=int, default=2,
201                         help="number of memory channels")
202     parser.add_argument("--mem-ranks", type=int, default=None,
203                         help="number of memory ranks per channel")
204     parser.add_argument("--mem-size", action="store", type=str,
205                         default="2GB",
206                         help="Specify the physical memory size")

```

Κομμάτι κώδικα από το config.json:

```

111     },
112     "cache line size": 64,

```

Κομμάτια κώδικα από το config.ini:

```

cache_line_size=64
eventq_index=0
exit_on_work_items=false
init_param=0

[system.voltage_domain]
type=VoltageDomain
eventq_index=0
voltage=3.3

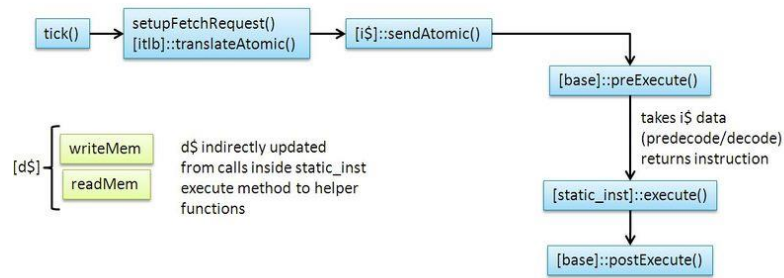
```

3. In-order CPU types supported by gem5

AtomicSimpleCPU:

Ο Atomic είναι επεξεργαστής που χρησιμοποιεί Atomic πρόσβαση στη μνήμη. Στον gem5, ο AtomicSimpleCPU εκτελεί όλες τις πράξεις για μια εντολή για κάθε CPU tick και μπορεί να έχει μια συνολική εκτίμηση του συνολικού χρόνου πρόσβασης στη μνήμη cache χρησιμοποιώντας τις εκτιμήσεις καθυστέρησης από τις ατομικές προσβάσεις. Φυσικά, ο AtomicSimpleCPU παρέχει την ταχύτερη λειτουργική προσομοίωση, και χρησιμοποιείται για fast-forwarding για να φτάσει σε περιοχή ενδιαφέροντος (ROI) στο gem5.

AtomicSimpleCPU



MinorCPU:

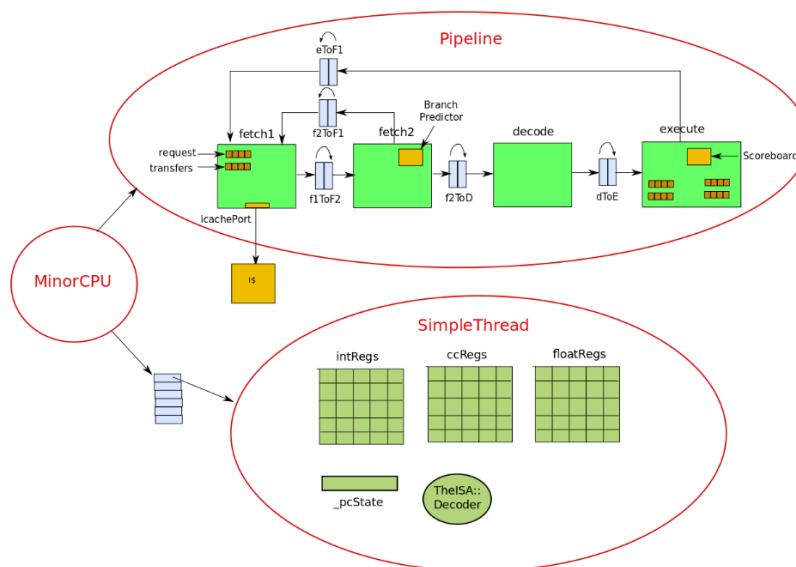
Πρόκειται για έναν επεξεργαστή με 4 στάδια pipelining. Τα τέσσερα στάδια είναι η fetch1, η fetch2, η decode και η execute. Η πρόσβαση ITLB και η λήψη της εντολής από την κύρια μνήμη γίνεται στο fetch1. Το fetch2 είναι υπεύθυνο για την αποκωδικοποίηση της εντολής, η decode είναι υπεύθυνη για book-keeping και η execute υλοποιεί το logic for issue, την εκτέλεση, τη μνήμη, το writeback και το commit. Όλα αυτά τα στάδια ορίζονται ως SimObjects στην κλάση Pipeline, η οποία υλοποιεί ολόκληρο το pipelining. Τα διαφορετικά στάδια του pipeline συνδέονται μεταξύ τους με Latches.

```

class Pipeline {
    /* Latches to connect the stages */
    Latch<ForwardLineData> f1ToF2;
    Latch<BranchData> f2ToF1;
    Latch<ForwardInstData> f2ToD;
    Latch<ForwardInstData> dToE;
    Latch<BranchData> eToF1;

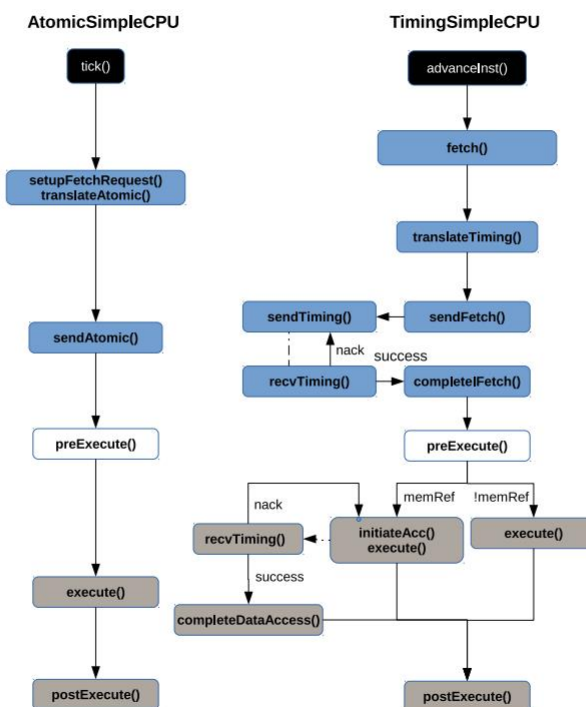
    /* Pipeline Stages */
    Execute execute;
    Decode decode;
    Fetch2 fetch2;
    Fetch1 fetch1;

    /* Action to be performed at each cycle (tick) */
    void evaluate();
}
  
```



TimingSimpleCPU:

Ο TimingSimpleCPU υιοθέτησε την πρόσβαση μνήμης χρονισμού αντί της Atomic. Αυτό σημαίνει ότι περιμένει, μέχρι η πρόσβαση στη μνήμη να επιστρέφει, πριν προχωρήσει, επομένως παρέχει κάποιο επίπεδο χρονισμού. Ο TimingSimpleCPU είναι επίσης γρήγορος στην εκτέλεση δεδομένου ότι απλοποιεί ορισμένες διαδικασίες, συμπεριλαμβανομένου του `pipelining`, πράγμα που σημαίνει ότι μόνο μία εντολή υπόκειται σε επεξεργασία οποιαδήποτε στιγμή. Κάθε αριθμητική εντολή εκτελείται από το TimingSimpleCPU σε έναν μόνο κύκλο, ενώ απαιτούνται πρόσβαση στη μνήμη πολλαπλούς κύκλους.



3.a.

TimingSimpleCPU:

```
----- Begin Simulation Statistics -----
final_tick                289462000
host_inst_rate            519226
host_mem_usage            665148
host_op_rate              612038
host_seconds              0.33
host_tick_rate            867410770
sim_freq                  1000000000000
sim_insts                 173192
sim_ops                   204225
sim_seconds               0.000289
sim_ticks                 289462000
system.cpu.Branches       36832
system.cpu.committedInsts 173192
system.cpu.committedOps   204225
```

```
# Number of ticks from beginning of simulation (restored from checkpoints and never reset)
# Simulator instruction rate (inst/s)
# Number of bytes of host memory used
# Simulator op (including micro ops) rate (op/s)
# Real time elapsed on the host
# Simulator tick rate (ticks/s)
# Frequency of simulated ticks
# Number of instructions simulated
# Number of ops (including micro ops) simulated
# Number of seconds simulated
# Number of ticks simulated
# Number of branches fetched
# Number of instructions committed
# Number of ops (including micro ops) committed
```

MinorCPU:

----- Begin Simulation Statistics -----			
final_tick	139169000		# Number of ticks from beginning of simulation (restored from checkpoints and never reset)
host_inst_rate	253424		# Simulator instruction rate (inst/s)
host_mem_usage	669504		# Number of bytes of host memory used
host_op_rate	300188		# Simulator op (including micro ops) rate (op/s)
host_seconds	0.69		# Real time elapsed on the host
host_tick_rate	202480944		# Simulator tick rate (ticks/s)
sim_freq	1000000000000		# Frequency of simulated ticks
sim_insts	174133		# Number of instructions simulated
sim_ops	206316		# Number of ops (including micro ops) simulated
sim_seconds	0.000139		# Number of seconds simulated
sim_ticks	139169000		# Number of ticks simulated
system.cpu.branchPred.BTBCorrect	0		# Number of correct BTB predictions (this stat may not work properly.
system.cpu.branchPred.BTBHitPct	47.298646		# BTB Hit Percentage
system.cpu.branchPred.BTBHits	15163		# Number of BTB hits
system.cpu.branchPred.BTBLookups	32058		# Number of BTB lookups
system.cpu.branchPred.RASInCorrect	7		# Number of incorrect RAS predictions.
system.cpu.branchPred.condInCorrect	1712		# Number of conditional branches incorrect
system.cpu.branchPred.condPredicted	26275		# Number of conditional branches predicted
system.cpu.branchPred.indirectHits	1795		# Number of indirect target hits.
system.cpu.branchPred.indirectLookups	2755		# Number of indirect predictor lookups.
system.cpu.branchPred.indirectMisses	960		# Number of indirect misses.
system.cpu.branchPred.lookups	39483		# Number of BP lookups
system.cpu.branchPred.usedRAS	4319		# Number of times the RAS was used to get a target.
system.cpu.branchPredIndirectMispredicted	283		# Number of mispredicted indirect branches.
system.cpu.committedInsts	174133		# Number of instructions committed
system.cpu.committedOps	206316		# Number of ops (including micro ops) committed
system.cpu.cpi	1.598422		# CPI: cycles per instruction
system.cpu.discardedOps	3568		# Number of ops (including micro ops) which were discarded before commit

3.b.

Χρησιμοποιώντας διαφορετικά μοντέλα CPU παρατηρούμε διαφορά στον αριθμό των ticks, στο instruction rate, στον χρόνο εκτέλεσης αλλά και στο tick rate. Αυτό συμβαίνει επειδή το pipelining σε κάθε μοντέλο είναι διαφορετικό και κατά συνέπεια αλλάζει ο αριθμός των εντολών που εκτελούνται σε κάθε κύκλο ρολογιού.

3.c.

Frequency 1GHz→2GHz

TimingSimpeCPU:

----- Begin Simulation Statistics -----			
final_tick	289462000		# Number of ticks from beginning of simulation (restored from checkpoints and never reset)
host_inst_rate	566336		# Simulator instruction rate (inst/s)
host_mem_usage	665152		# Number of bytes of host memory used
host_op_rate	667535		# Simulator op (including micro ops) rate (op/s)
host_seconds	0.31		# Real time elapsed on the host
host_tick_rate	946051945		# Simulator tick rate (ticks/s)
sim_freq	1000000000000		# Frequency of simulated ticks
sim_insts	173192		# Number of instructions simulated
sim_ops	204225		# Number of ops (including micro ops) simulated
sim_seconds	0.000289		# Number of seconds simulated
sim_ticks	289462000		# Number of ticks simulated
system.cpu.Branches	36832		# Number of branches fetched
system.cpu.committedInsts	173192		# Number of instructions committed
system.cpu.committedOps	204225		# Number of ops (including micro ops) committed

MinorCPU:

----- Begin Simulation Statistics -----			
final_tick	139169000		# Number of ticks from beginning of simulation (restored from checkpoints and never reset)
host_inst_rate	266712		# Simulator instruction rate (inst/s)
host_mem_usage	669500		# Number of bytes of host memory used
host_op_rate	315937		# Simulator op (including micro ops) rate (op/s)
host_seconds	0.65		# Real time elapsed on the host
host_tick_rate	213104106		# Simulator tick rate (ticks/s)
sim_freq	1000000000000		# Frequency of simulated ticks
sim_insts	174133		# Number of instructions simulated
sim_ops	206316		# Number of ops (including micro ops) simulated
sim_seconds	0.000139		# Number of seconds simulated
sim_ticks	139169000		# Number of ticks simulated
system.cpu.branchPred.BTBCorrect	0		# Number of correct BTB predictions (this stat may not work properly.
system.cpu.branchPred.BTBHitPct	47.298646		# BTB Hit Percentage
system.cpu.branchPred.BTBHits	15163		# Number of BTB hits
system.cpu.branchPred.BTBLookups	32058		# Number of BTB lookups
system.cpu.branchPred.RASInCorrect	7		# Number of incorrect RAS predictions.
system.cpu.branchPred.condInCorrect	1712		# Number of conditional branches incorrect
system.cpu.branchPred.condPredicted	26275		# Number of conditional branches predicted
system.cpu.branchPred.indirectHits	1795		# Number of indirect target hits.
system.cpu.branchPred.indirectLookups	2755		# Number of indirect predictor lookups.
system.cpu.branchPred.indirectMisses	960		# Number of indirect misses.
system.cpu.branchPred.lookups	39483		# Number of BP lookups
system.cpu.branchPred.usedRAS	4319		# Number of times the RAS was used to get a target.
system.cpu.branchPredIndirectMispredicted	283		# Number of mispredicted indirect branches.
system.cpu.committedInsts	174133		# Number of instructions committed
system.cpu.committedOps	206316		# Number of ops (including micro ops) committed
system.cpu.cpi	1.598422		# CPI: cycles per instruction
system.cpu.discardedOps	3568		# Number of ops (including micro ops) which were discarded before commit

Πηγές:

- http://www.gem5.org/Main_Page
- <https://nitish2112.github.io/post/gem5-minor-cpu/>
- <http://www.m5sim.org/SimpleCPU>
- https://raw.githubusercontent.com/arm-university/arm-gem5-rsk/master/gem5_rsk.pdf

Github Link: https://github.com/antomavr/gem5_lab1