

## ΠΑΡΑΔΟΤΕΟ 1ου ΕΡΓΑΣΤΗΡΙΟΥ (Πρώτο Μέρος)

**Μαυρομανώλης Αντώνιος**

**A.M. 9010**

[antomavr@ece.auth.gr](mailto:antomavr@ece.auth.gr)

1. Από το αρχείο `starter_se.py` οι αρχικές παράμετροι που έχουν περάσει στον `gem5` για το σύστημα προς εξομίωση είναι:

- **Cache line size:** 64
- **Voltage:** "3.3V"
- **CPU model to use:** "minor" (κατά την εκτέλεση της εντολής (\$ ./build/ARM/gem5.opt -d hello\_result configs/example/arm/starter\_se.py --cpu="minor" "tests/test-progs/hello/bin/arm/linux/hello") υπάρχει το flag --cpu="minor".)
- **CPU frequency :** 1GHz"

Default:

- **Number of CPU cores:** 1
- **Type of memory to use:** "DDR3\_1600\_8x8"
- **Number of memory channels:** 2
- **Number of memory ranks per channel:** None
- **Specify the physical memory size:** "2GB"

2. Ανοίγοντας τα αρχεία config.json και config.ini επαληθεύεται η απάντηση του πρώτου ερωτήματος, καθώς τα στοιχεία αντιστοιχούν μεταξύ τους.

Κομμάτι κώδικα από το starter se.py:

[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

```

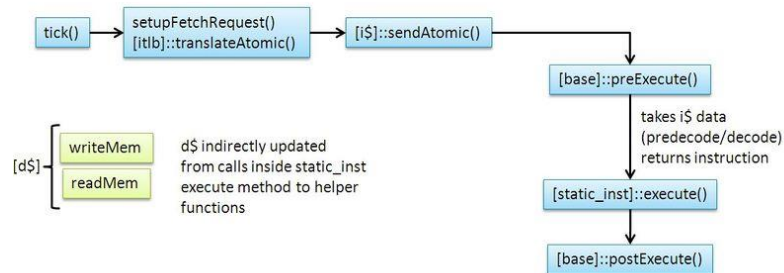
66 # Pre-defined CPU configurations. Each tuple must be ordered as : (cpu_class,
67 # ll_icache_class, ll_dcache_class, walk_cache_class, l2_Cache_class). Any of
68 # the cache class may be 'None' if the particular cache is not present.
69 cpu_types = {
70     "atomic" : ( AtomicSimpleCPU, None, None, None, None),
71     "minor" : (MinorCPU,
72               devices.L1I, devices.L1D,
73               devices.WalkCache,
74               devices.L2),
75     "hpi" : ( HPI.HPI,
76             HPI.HPI_ICache, HPI.HPI_DCache,
77             HPI.HPI_WalkCache,
78             HPI.HPI_L2)
79 }

```

### atomic:

Ο Atomic είναι επεξεργαστής που χρησιμοποιεί ατομική πρόσβαση στη μνήμη. Χρησιμοποιεί τα latency estimates από τις ατομικές προσβάσεις για να υπολογίσει το συνολικό χρόνο πρόσβασης στη κρυφή μνήμη (cache). Ο Atomic CPU προέρχεται από τον BaseSimpleCPU και υλοποιεί λειτουργίες ανάγνωσης και εγγραφής στη μνήμη, καθώς επίσης και tick (ορίζει δηλαδή το τι συμβαίνει σε κάθε κύκλο του ρολογιού). Ορίζει τη θύρα που χρησιμοποιείται για να συνδέσει τη μνήμη και συνδέει τη CPU με την cache.

## AtomicSimpleCPU



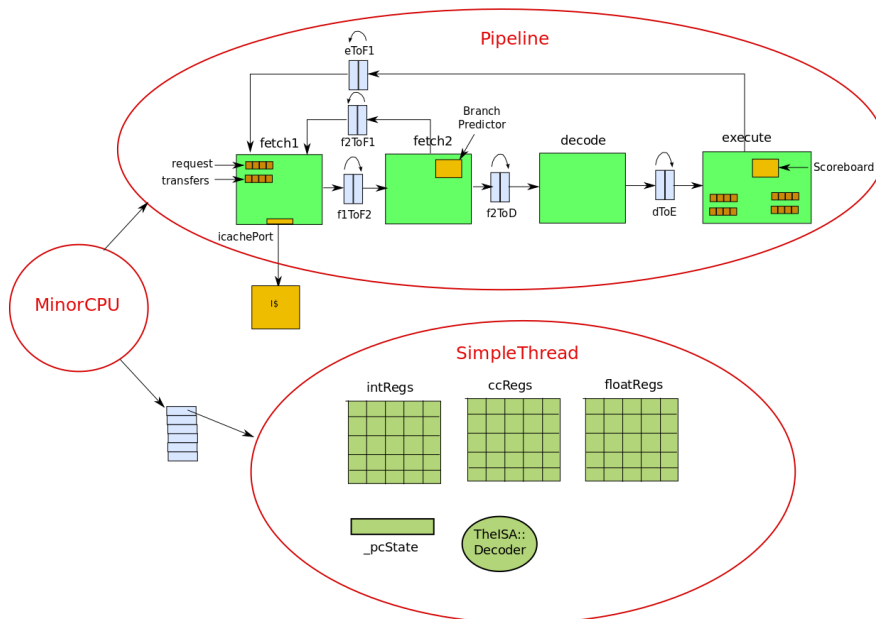
### minor:

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;

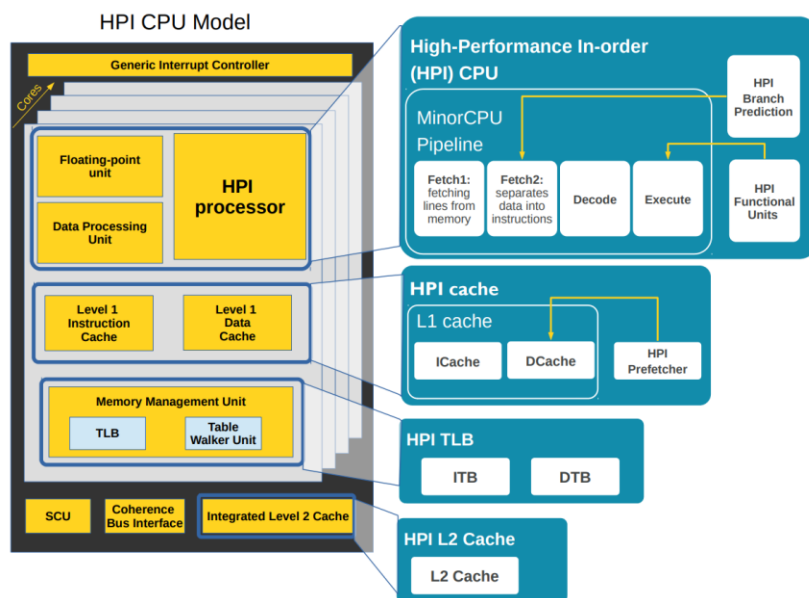
    /* Pipeline Stages */
    Execute execute;
    Decode decode;
    Fetch2 fetch2;
    Fetch1 fetch1

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



### hpi:

Ο HPI επεξεργαστής είναι βασισμένος στην αρχιτεκτονική βραχίονα. Το μοντέλο χρονισμού του HPI επεξεργαστή αντιπροσωπεύει μια πραγματική in-order υλοποίηση του Armv8-A. Ο pipeline του HPI CPU χρησιμοποιεί το ίδιο μοντέλο τεσσάρων σταδίων με το Minor CPU.



*Building blocks of the Arm High-performance In-order CPU in gem5*

### 3.a.

#### Atomic:

final_tick and never reset)	57613500	# Number of ticks from beginning of simulation (restored from checkpoints
host_inst_rate	861734	# Simulator instruction rate (inst/s)
host_mem_usage	2234256	# Number of bytes of host memory used
host_op_rate	1020454	# Simulator op (including micro ops) rate (op/s)
host_seconds	0.20	# Real time elapsed on the host
host_tick_rate	284668557	# Simulator tick rate (ticks/s)
sim_freq	1000000000000	# Frequency of simulated ticks

#### Minor:

final_tick and never reset)	79766750	# Number of ticks from beginning of simulation (restored from checkpoints
host_inst_rate	266605	# Simulator instruction rate (inst/s)
host_mem_usage	2248588	# Number of bytes of host memory used
host_op_rate	315793	# Simulator op (including micro ops) rate (op/s)
host_seconds	0.65	# Real time elapsed on the host
host_tick_rate	121852903	# Simulator tick rate (ticks/s)
sim_freq	1000000000000	# Frequency of simulated ticks

#### HPI:

final_tick and never reset)	86996750	# Number of ticks from beginning of simulation (restored from checkpoints
host_inst_rate	233762	# Simulator instruction rate (inst/s)
host_mem_usage	2253784	# Number of bytes of host memory used
host_op_rate	276899	# Simulator op (including micro ops) rate (op/s)
host_seconds	0.75	# Real time elapsed on the host
host_tick_rate	116530526	# Simulator tick rate (ticks/s)
sim_freq	1000000000000	# Frequency of simulated ticks

### 3.b.

Χρησιμοποιώντας διαφορετικά μοντέλα CPU παρατηρούμε διαφορά 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

#### Atomic:

final_tick and never reset)	57613500	# Number of ticks from beginning of simulation (restored from checkpoints
host_inst_rate	829289	# Simulator instruction rate (inst/s)
host_mem_usage	2234256	# Number of bytes of host memory used
host_op_rate	982092	# Simulator op (including micro ops) rate (op/s)
host_seconds	0.21	# Real time elapsed on the host
host_tick_rate	273967359	# Simulator tick rate (ticks/s)

## Minor:

final_tick and never reset)	76878250	# Number of ticks from beginning of simulation (restored from checkpoints)
host_inst_rate	276708	# Simulator instruction rate (inst/s)
host_mem_usage	2248588	# Number of bytes of host memory used
host_op_rate	327764	# Simulator op (including micro ops) rate (op/s)
host_seconds	0.63	# Real time elapsed on the host
host_tick_rate	121889531	# Simulator tick rate (ticks/s)

## HPI:

final_tick and never reset)	83922250	# Number of ticks from beginning of simulation (restored from checkpoints)
host_inst_rate	237445	# Simulator instruction rate (inst/s)
host_mem_usage	2253784	# Number of bytes of host memory used
host_op_rate	281263	# Simulator op (including micro ops) rate (op/s)
host_seconds	0.73	# Real time elapsed on the host
host_tick_rate	114183552	# Simulator tick rate (ticks/s)

## Timing→ O3CPU

## Atomic:

final_tick and never reset)	57613500	# Number of ticks from beginning of simulation (restored from checkpoints)
host_inst_rate	833241	# Simulator instruction rate (inst/s)
host_mem_usage	2234256	# Number of bytes of host memory used
host_op_rate	985636	# Simulator op (including micro ops) rate (op/s)
host_seconds	0.21	# Real time elapsed on the host
host_tick_rate	274947159	# Simulator tick rate (ticks/s)

## Minor:

final_tick and never reset)	3008928500	# Number of ticks from beginning of simulation (restored from checkpoints)
host_inst_rate	152643	# Simulator instruction rate (inst/s)
host_mem_usage	2239636	# Number of bytes of host memory used
host_op_rate	180821	# Simulator op (including micro ops) rate (op/s)
host_seconds	1.14	# Real time elapsed on the host
host_tick_rate	2631983001	# Simulator tick rate (ticks/s)

## HPI:

final_tick and never reset)	3335975500	# Number of ticks from beginning of simulation (restored from checkpoints)
host_inst_rate	138021	# Simulator instruction rate (inst/s)
host_mem_usage	2243480	# Number of bytes of host memory used
host_op_rate	163500	# Simulator op (including micro ops) rate (op/s)
host_seconds	1.26	# Real time elapsed on the host
host_tick_rate	2638538736	# Simulator tick rate (ticks/s)

**Πηγές:**

- [http://www.gem5.org/Main\\_Page](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](https://raw.githubusercontent.com/arm-university/arm-gem5-rsk/master/gem5_rsk.pdf)