INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

CS5119- Advanced Computer Architecture Lab

Subham Ball(152202017)

September 25, 2022

LAB 4

(1) Compare the performance of AtomicSimpleCPU and TimingSimpleCPU models using default configuration script given in config->example. You can use hello world binary inside tests- >test-progs->hello->bin->riscv->hello. Discuss the major differences you observe in the m5out->stats.txt for both models.

Solution:
TimingSimpleCPU: it is a version of simpleCPU that use for timing memory accesses.It stalls on cache accesses and waits for the memory system to respond prior to proceeding.

AtomicSimpleCPU: it is the version of SimpleCpu that uses atomic memory accesses and it is derived from BasicSimpleCPU.it uses the latency estimates from the atomic access to estimate overall cache access time.

Major differences of obeservation in the m5out->stats.txt for both models		
Parameter	AtomicSimpleCPU	TimingSimpleCPU
simSeconds	0.000006	0.0004553
simTick	5943000	4546460003
hostSeconds	0.02	0.033
CPU Cycles	11887	909292
Number of busy cycles	11886.998000	909291.9980003
priority MinLatency	0.000000000000	0.000000018750
priorityMaxLatency	0.000000000000	0.00052264675
Bus in read state	0	1746
Bus in write state	0	9
Read row Hit rate	0	89.81
Write Row Hit rate	0	90.62
Data bus utilization	0	9.13
Data bus utilization for Read	0	9.03
Data bus utilization for Write	0	0.11
page hit rate	nan	89.82
Bus in read state	16189	17463
total Energy	2282400	278973615
average Power	384.048460	613.60622
average read bandwidth	0	143969154.02313009
average write bandwidth	0	15748516.42816609
number of branch		1306

Observation : Simulation time of AtomicSimpleCPU is take less than TimingSimpleCPU.
Reason : AtomicSimpleCPU memmory requests are finish immediately but TimingSimpleCPU take time to go through the memory and return.

(2)Repeat task A, but this time compare riscv and x86 based on TimingSimpleCPU model. Please highlight the differences you observe.

Solution:

Major differences of obeservation in riscv and X86 based on TimingSimple CPU		
Parameter	riscv	x86
simSeconds	0.000422	0.000455
simTick	421689000	4546460000
hostSeconds	0.02	0.033
CPU Cycles	843378	909292
Number of busy cycles	843377.998000	909291.9980003
number of branches	1224	1306
priority MinLatency	0.000000018750	0.000000018750
priorityMaxLatency	0.000510342750	0.00052264675
Avg BUS latency	5000.00	5000.00
Avg memory latency	22517.50	22311.26
Read row Hit rate	89.29	89.81
Write Row Hit rate	90.18	90.62
Data bus utilization	9.06	9.13
Data bus utilization for Read	8.93	9.03
Data bus utilization for Write	0.13	0.11
page hit rate	89.31	89.82
Bus in read state	16189	17463
Bus in write state	105	93
Read queue length	1	1
Write queue length	24.43	23.46
total Energy	280973295	278973615
average Power	666.304540	613.60622
average read bandwidth	79183948.36004734	143969154.02313009
average write bandwidth	18762642.61102377	15748516.42816609

Observation : As we observe from the simulation data, for some cases riscv is better than x86 and take less time or less cycle to complete the task(simSeconds, CPU cycle etc.) but some cases x86 are better than riscv(read write hit rate, avg memory latency etc.)

[a small Research: RISC-V has a option for weak memory model and memory model is very importent for parallel computing. Weaker memory models enable parallel computing to be implemented by CPU designers with less silicon and fewer stages, making them quicker and more power efficient.]