Project 1 Report: Bank-level Parallelism and Memory Controller Design

Tai Duc Nguyen
ECEC 623: Advanced Topics in Computer Architecture

February 11, 2020

Abstract

While Morse's Law has been true for modern processors, it does not apply to memory. Hence, in order for the memory speed to keep up with processing speed, many parallelism has to be exploited at many levels inside the memory of a von-Neumann computer. In this project, the effect of bank-level parallelism is explored along with 2 elementary memory controller design: FCFS - First Come First Serve, and FRFCFS - First Ready, First Come First Serve.

1 Experimental Setup

In order to see how bank-level parallelism can improve memory speed, it is necessary to lay out a few constraints and metrics. The simulation assumes the following (but may not be limited to):

1. DRAM Memory timings are:
   (a) Reading takes 53 clock cycles
   (b) Writing takes 53 clock cycles
2. PCM Memory timings are:
   (a) Reading takes 57 clock cycles
   (b) Writing takes 162 clock cycles
3. Maximum number of requests in the waiting queue is: 64
4. The number of banks being tested are: 2, 4, 8, 16
5. FR-FCFS loop searching for new request when the first one has bank-conflict can be done in 1 clock cycle
6. A bank conflict of a memory request only counts once even though it could be queued and conflicted again in the next clock cycle

From the assumptions above a simulation written in C is created. It first read a memory request from a memory trace, send such request to the "Waiting" queue. Once a request is in the queue, if the bank it targets is not busy, it will be pushed to the "Pending" queue and served in the next clock cycle. However, if the bank is busy, such request must wait until the bank is ready (in FCFS), or the program will find another request in the "Waiting" queue which targets a readied-bank.

The performance of bank-parallelism is then tested with FCFS & FRFCFS controllers, and DRAM &

PCM Memory types. The metrics includes:

1. Average Access Latency (**AAL**)
2. Number of Bank Conflicts (**BCF**)
3. Total Execution Time (**EXE**)

The 5 memory traces in this experiments are extracted from the following benchmarks: **503.bwaves_r, 538.imagick_r, 554.roms_r, 557.xz_r, and MP10**. The specifications for these benchmarks could be found in `https://www.spec.org/cpu2017/Docs/benchmarks/`

2 Results and Discussion

This section details the results from the simulation with graphs.

503.bwaves_r.trace.processed				FRFCFS			FCFS					
nbanks	clk_r	clk_w	nreqs	AAL	BCF	EXE	AAL	BCF	EXE	AAL (FCFS/FRFCFS)	BCF (FCFS/FRFCFS)	EXE (FCFS/FRFCFS)
2	53	53	58140705	1696	29077602	1541039941	2651	45234346	2408793013	1.563	1.556	1.563
4	53	53	58140705	850	14978598	772493782	2355	40111363	2139823793	2.771	2.678	2.770
8	53	53	58140705	468	9021730	425213480	2205	37516677	2003705063	4.712	4.158	4.712
16	53	53	58140705	321	6502837	292455486	2120	36044766	1926215754	6.604	5.543	6.586
2	57	162	58140705	3433	29468107	3119504982	5179	48932457	4705166590	1.509	1.661	1.508
4	57	162	58140705	1721	15429823	1563831220	4380	44502444	3979017558	2.545	2.884	2.544
8	57	162	58140705	943	9493160	856789468	3945	41769742	3584644042	4.183	4.400	4.184
16	57	162	58140705	642	6920997	583378847	3681	39974240	3344119140	5.734	5.776	5.732

Figure 1: Trace 503's simulation results, including improvement ratios between metrics of FCFS and FRFCFS

538.imagick_r.trace.processed				FRFCFS			FCFS					
nbanks	**clk_r**	**clk_w**	**nreqs**	AAL	BCF	EXE	AAL	BCF	EXE	AAL (FCFS/FRFCFS)	BCF (FCFS/FRFCFS)	EXE (FCFS/FRFCFS)
2	53	53	1263760	1696	631921	33491868	2247	833210	44381810	1.325	1.319	1.325
4	53	53	1263760	848	321780	16746793	1595	614748	31506787	1.881	1.910	1.881
8	53	53	1263760	424	176949	8378127	1059	420427	20916912	2.498	2.376	2.497
16	53	53	1263760	212	127898	4202657	634	259824	12526508	2.991	2.031	2.981
2	57	162	1263760	1824	631921	36019556	2416	833210	47714654	1.325	1.319	1.325
4	57	162	1263760	912	321780	18010701	1714	614748	33857543	1.879	1.910	1.880
8	57	162	1263760	456	176949	9010435	1137	420427	22457492	2.493	2.376	2.492
16	57	162	1263760	228	127981	4519773	679	259858	13423580	2.978	2.030	2.970

Figure 2: Trace 538's simulation results, including improvement ratios between metrics of FCFS and FRFCFS

554.roms_r.trace.processed				FRFCFS			FCFS					
nbanks	clk_r	clk_w	nreqs	AAL	BCF	EXE	AAL	BCF	EXE	AAL (FCFS/FRFCFS)	BCF (FCFS/FRFCFS)	EXE (FCFS/FRFCFS)
2	53	53	117486447	1697	58812821	3117026017	2580	88885129	4737990343	1.520	1.511	1.520
4	53	53	117486447	882	31202310	1620726456	2387	82097562	4383687591	2.706	2.631	2.705
8	53	53	117486447	486	17766503	892399211	2278	78260224	4183375182	4.687	4.405	4.688
16	53	53	117486447	305	13225751	561535869	2242	76965338	4115754735	7.351	5.819	7.329
2	57	162	117486447	3483	59664234	6393984324	5047	98111518	9265607009	1.449	1.644	1.449
4	57	162	117486447	1811	31956950	3324765362	4253	87694562	7807515430	2.348	2.744	2.348
8	57	162	117486447	996	18352094	1828719301	3912	83308644	7182463673	3.928	4.539	3.928
16	57	162	117486447	620	14744650	1139390855	3790	81601737	6959079275	6.113	5.534	6.108

Figure 3: Trace 554's simulation results, including improvement ratios between metrics of FCFS and FRFCFS

557.xz_r.trace.processed				FRFCFS			FCFS					
nbanks	clk_r	clk_w	nreqs	AAL	BCF	EXE	AAL	BCF	EXE	AAL (FCFS/FRFCFS)	BCF (FCFS/FRFCFS)	EXE (FCFS/FRFCFS)
2	53	53	2021346	1696	1010958	53580517	1970	1168362	62237290	1.162	1.156	1.162
4	53	53	2021346	848	514234	26809992	1291	806619	40788393	1.522	1.569	1.521
8	53	53	2021346	425	265972	13427106	919	581396	29040074	2.162	2.186	2.163
16	53	53	2021346	213	138580	6745825	712	455490	22501109	3.343	3.287	3.336
2	57	162	2021346	2127	1014220	67181490	2484	1225596	78467152	1.168	1.208	1.168
4	57	162	2021346	1064	518192	33634374	1550	830621	48983768	1.457	1.603	1.456
8	57	162	2021346	534	267142	16872572	1087	596918	34333816	2.036	2.234	2.035
16	57	162	2021346	269	140106	8507480	827	466037	26122806	3.074	3.326	3.071

Figure 4: Trace 557's simulation results, including improvement ratios between metrics of FCFS and FRFCFS

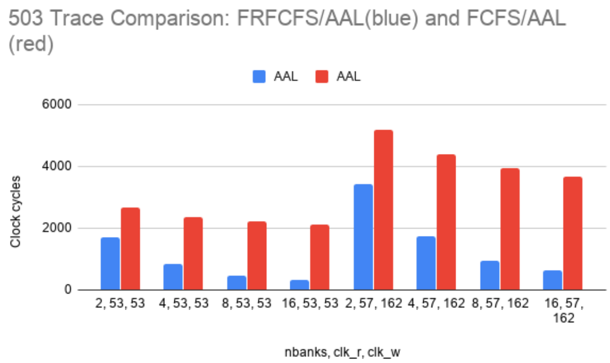
MP10.trace.processed				FRFCFS			FCFS					
nbanks	clk_r	clk_w	nreqs	AAL	BCF	EXE	AAL	BCF	EXE	AAL (FCFS/FRFCFS)	BCF (FCFS/FRFCFS)	EXE (FCFS/FRFCFS)
2	53	53	23421913	1696	11715451	620849588	2557	17558834	936081911	1.508	1.499	1.508
4	53	53	23421913	877	6167363	321128831	2406	16500504	880813034	2.743	2.675	2.743
8	53	53	23421913	478	3471735	174946914	2335	15997836	854642453	4.885	4.608	4.885
16	53	53	23421913	298	2528555	109226054	2315	15854213	847220967	7.768	6.270	7.757
2	57	162	23421913	3374	11865758	1234876429	4859	19491434	1778362910	1.440	1.643	1.440
4	57	162	23421913	1752	6266615	641289909	4083	17183011	1494315782	2.330	2.742	2.330
8	57	162	23421913	959	3580982	351222466	3861	16535026	1413252876	4.026	4.617	4.024
16	57	162	23421913	595	2821078	217926999	3797	16323585	1389641953	6.382	5.786	6.377

Figure 5: Trace MP10's simulation results, including improvement ratios between metrics of FCFS and FRFCFS

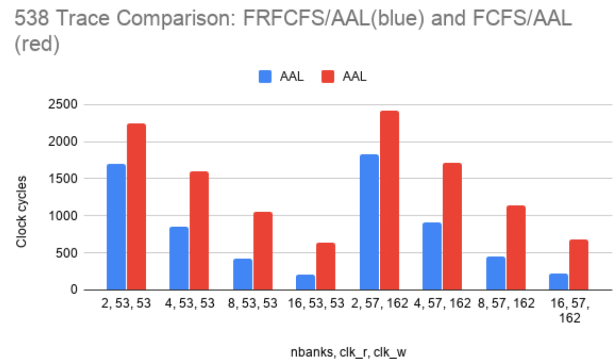
From these results, there are a few trends which emerge:

1. AAL only slightly decreases with increasing number of banks for trace 503, 554 and MP10. This signifies that these programs are mostly trying to read or write to the same memory address (maybe in a loop). The memory accesses are not alocated equally across the banks.
2. AAL decreases a long with increasing number of banks for trace 538 and 557. This tells that these 2 benchmarks includes memory accesses which are spread equally across the banks.
3. PCM memory is much slower on average than DRAM memory.
4. The higher the number of banks, the more significant of improvements seen in FRFCFS vs FCFS. In cases where there are 16 banks, the improvement is as large as 7.8 times. With a lower number of banks, the improvement is around 1.5 to 2 times.
5. FRFCFS provide such an improvement over FCFS because it keeps the Pending queue full most of the time, hence, saving many clock cycles by not stalling until the conflict is resolved.
6. AAL, BCF and EXE are all directly proportional to one another.

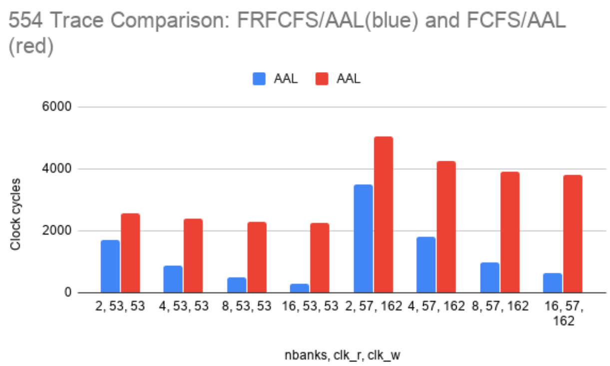
Graphical results are plotted on the next page.



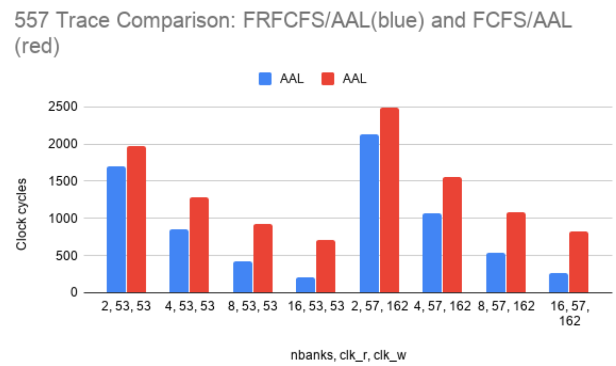
(a) Trace 503



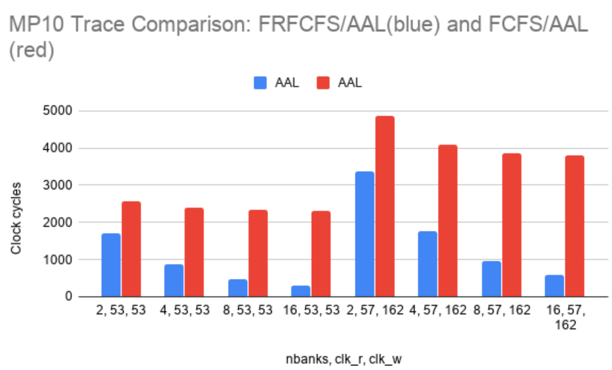
(b) Trace 538



(c) Trace 554



(d) Trace 557



(e) Trace MP10

Figure 6: Graph summary of simulations' results