

# Memory performance of ARM processors and its relevance to High Energy Physics

**T Wrigley, G Harmsen and B Mellado**

University of the Witwatersrand, 1 Jan Smuts Avenue, Johannesburg, South Africa 2000

E-mail: thomas.wrigley@cern.ch

**Abstract.** Projects such as the to-be upgraded ATLAS detector at the Large Hadron Collider at CERN are expected to produce data in volumes which far exceed current system data throughput capacities. In addition, cost considerations for large-scale computing systems remain a source of general concern. A potential solution involves using low-cost, low-power ARM processors in large arrays in a manner which provides massive parallelisation and high rates of data throughput (relative to existing large-scale computing designs). Giving greater priority to both throughput-rate and cost considerations increases the relevance of primary memory performance and design optimisations to overall system performance. Using several primary memory performance benchmarks to evaluate various aspects of RAM and cache performance, we provide characterisations of the performances of three different models of ARM-based SoC, namely the Cortex-A9, Cortex-A7 and Cortex-A15. We then discuss the relevance of these results to high throughput-rate computing and the potential for ARM processors. Finally, applications to the upgrade of the on-line and off-line data processing at the ATLAS detector are also discussed.

## 1. Introduction and Background

The term Big Data is widespread and its usage appears to be continually increasing [1]. Very large-scale scientific experiments such as the Large Hadron Collider (LHC) at CERN or the under construction Square Kilometre Array (SKA) telescope are increasingly being referred to as belonging to the realm of Big Science. A key challenge of large-scale scientific experiments is the vast amounts of data that they generate requiring far greater data throughput capacity and much larger data storage capacity than existing technologies and infrastructure can feasibly or reasonably accommodate. The ATLAS detector, one of seven within the LHC, makes use of a multi-level triggering system to filter output data, due to *post hoc* processing and long term storage limitations. Even after filtering, the amount of data currently committed to storage is very large and the scheduled upgrade to the LHC will further exacerbate this problem by a significant degree.

A concept known as Data Stream Computing has been proposed as a potential means of addressing these high data throughput challenges [2]. Data Stream Computing (DSC), briefly, has the following three characteristics: very high data throughput rates, severely limited or no offline storage of data, and programming simplicity (i.e. common, easily programmable architectures as opposed to specialised, custom architectures). In addition to the characteristics of DSC, affordability and energy efficiency are two major concerns for large-scale computing in

general. A potential solution involves the use of ARM processors, which are low-power, low-cost and low-energy consumption systems-on-chip (SoC), in large arrays which would provide very high levels of parallelisation. ARM-based SoCs, which are commonly used in mobile devices such as smartphones and tablets, are low-cost, mass-produced and potentially highly energy-efficient [3], all of which bodes well for both affordability and energy efficiency.

Although large scale computing has traditionally placed its primary focus on processor performance, the relevance of memory performance to overall system performance is being increasingly widely acknowledged [4, 5]. Memory performance is a key component of overall system performance and is particularly important for throughput rates, memory bottlenecks could potentially affect energy-efficiency and cost through under-utilisation of existing system hardware. Using ARM-based SoCs in any proposed solution therefore requires that the performance of ARM-based SoCs be properly characterised and understood. Memory performance of existing ARM-based systems and its relevance to DSC for large-scale scientific experiments, particularly ATLAS, is the focus of this paper.

## 2. Experimental Configuration

The performance of three models of development board containing ARM-based systems-on-chip was evaluated. For practical and financial reasons, commercially available development boards containing ARM-based SoCs were used for the purposes of benchmarking. The technical specifications of these boards are listed in table 1 below.

**Table 1.** Development board hardware & OS specifications

	Cortex-A7	Cortex-A9	Cortex-A15
Platform	Cubieboard2	Wandboard Quad	Odroid-XU+E
SoC	Allwinner A20	Freescale i.MX6Q	Samsung Exynos 5410
Cores	2	4	4 (+ 4 Cortex-A7)
Max. CPU Clock (MHz)	1008	996	1600
L1 Cache (kB)	32	32	32
L2 Cache (kB)	256	1024	2048
RAM Size (MB)	1024	2048	2048
DDR3 RAM Type	432 MHz 32 bit	528 MHz 64 bit	800 MHz 64 bit
2014 Price (USD)	65	129	169
OS	Ubuntu	Linaro	Ubuntu

A Linux-based distribution was installed on all three types of boards. Three benchmarking software programmes were used to evaluate the memory performance of these three boards, namely the LMBench benchmark suite, the STREAM benchmark and the Parallel Memory Bandwidth Benchmark (*pmbw*).

The LMBench benchmarking suite analyses several aspects of memory performance this study focuses on the measures of memory latency. The STREAM benchmark provides a measure of sustained memory bandwidth. The benchmark works by generating an array of random numbers of a specified size (which is then stored in RAM) and performs four types of operations, namely copy, scale, add and triad. Measures of sustained bandwidth are then produced for each of these four tests. The *pmbw* benchmark is similar to STREAM in that it also provides a measure of sustained memory bandwidth, but does so by means of 14 separate

subtests, each performing a slightly different operation. These variations are: sequential scanning or a random access (permutation walking) test, write or read operation, bit size transferred in each operation, pointer-based iterations vs index-based array access, and number of operations per loop (1 - Simple vs 16 Unroll) [6]. Two of the subtests involve Multiroll Loops and are not analysed here. The benchmark is designed to automatically detect the amount of physical RAM available. It then generates an array and runs one of the subtest routines. The allocated array size is then increased and the subtest routine is then repeated. This is repeated until the highest power of 2 able to fit onto the systems RAM is reached. These steps are repeated for each one of the subtest routines. *pmbw* is useful because it allows both for comparisons to STREAM and will potentially yield deeper insight into memory performance.

### 3. Results and Discussion

#### 3.1. STREAM and LMBench

For the STREAM benchmark, which measures sustained memory bandwidth, the Cortex-A15 is clearly shown to be the best-performing of the three systems, both in terms of absolute bandwidth and bandwidth efficiency (i.e. percentage of theoretical maximum obtained). The Cortex-A7 displays reasonable bandwidth efficiency, while the Cortex-A9, which is the oldest of the three systems, achieves very low bandwidth efficiency, only reaching 16% of its theoretical maximum. In the case of RAM and cache latencies, the Cortex-A7 and Cortex-A15 both perform well, recording low latencies. The performance of the Cortex-A9 in this regard is also inferior to the A7 and A15 SoCs. For both of these benchmarks, a clear positive correlation can be seen between age of SoC design and performance. Table 2 below summarises the results obtained from both LMBench and STREAM for all three boards.

**Table 2.** Development board hardware & OS specifications

	Cortex-A7	Cortex-A9	Cortex-A15
Copy (MB/s)	1996	1329	6066
Scale (MB/s)	1444	1110	6114
Add (MB/s)	757	1448	5413
Triad (MB/s)	702	1290	5275
RAM (Theoretical MB/s)	3296	8054	12207
RAM BW Efficiency (%)	37	16	47
L1 Latency (ns)	3.02	4.02	2.51
L2 Latency (ns)	9.2	30.8	13.8
RAM Latency (ns)	58.5	119.8	104.8

#### 3.2. *pmbw*

The design of the *pmbw* benchmark means that each subtest routine generates several hundred sets of observations between 200 and 300 observations in the case of the three systems tested here. Because there are several hundred observations per subtest and 12 subtests which are analysed here, the volume of data produced by this benchmark for each system is very large numbering around several thousand observations. For this reason, statistical tools are useful for extracting meaning from these data sets. A statistical test known as analysis of variance (ANOVA) was used for primary analysis of the results of this benchmark. ANOVA is used to compare multiple datasets and determine whether the individual means of these datasets

are equal to one another. More specifically, ANOVA compares the variance within each of these datasets to the variance which is present between these datasets and determines whether statistically significant differences exist between these datasets [7]. If statistically significant differences between these datasets do exist, various *post hoc* tests and analyses can then be used to gain greater insight into the distribution and nature of these differences.

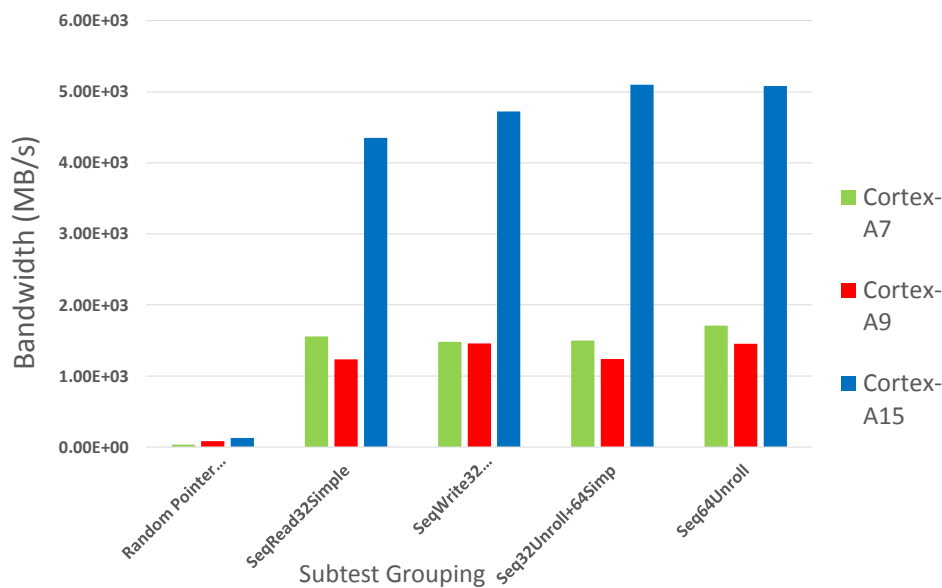
In this case, each subtest (with its 200-300 observations per system) represents a dataset and ANOVA is used to determine whether these individual subtests are statistically similar to one another. A two-way analysis of variance showed that significant differences existed between the subtest groups for all three boards i.e. at least one pair of means was different from one another. Post hoc analysis was then conducted to gain greater insight into the nature and distribution of these results. This analysis revealed the results generated by the 12 subtests appear to be distributed into five general groupings, with each grouping being made up of two, three or four subtests. As each subtest has 5 primary characteristics which vary, the existence of these five groupings gives a greater level of insight into which of these characteristics appear to have the greatest impact on performance insights which allow for memory performance to be better understood. The types of subtests which make up each grouping are briefly detailed in Table 3 below.

**Table 3.** Constituent subtest types of *pmbw* result groupings

Group No.	Subtest types	Number of subtests in group
1	Random Pointer Permutations (Perm)	2
2	Sequential Reading 32 bit Simple Loop	2
3	Sequential Write 32 bit Simple & Unroll Loop	3
4	Sequential 32 bit Unroll & 64 bit Simple Loop	3
5	Sequential 64 bit Unroll Loop	2

Based on the subtest result groupings determined above, the average of the two/three/four RAM bandwidth results for each of the five groupings was plotted. These bandwidth results are shown in figure 1 below. The first grouping (Random Pointer Permutation) is substantially lower than the other four groupings. This is, however, consistent with expectations, as this benchmark is based on a random pointer permutation and is essentially a measure of raw bandwidth and latency for one memory fetch cycle, while the other four are measures of sustained memory bandwidth for sequential scanning [6]. These results indicate that the Cortex-A7 achieves the lowest performance (approx.. 35 MB/s), the Cortex-A9 produces more than double that rates (approx. 85 MB/s) and the Cortex-A15 is again the best performer (approx.. 127 MB/s). This appears to be inconsistent with the memory latency and sustained memory bandwidth results obtained by LMBench and STREAM, which showed that the newer Cortex-A15 was the best performing of the system, the Cortex-A7 the second best performing and that the Cortex-A9 was the worst performing system by a significant margin. While these two random pointer permutation subtests are not solely dependent on memory latency, this would be expected to have some effect on random memory access performance. It is not immediately clear why the results produced by *pmbw* appear to conflict with the trends implied by the obtained LMBench results, although factors such as the Cortex-A9 SoCs 64 bit RAM bus width compared to the Cortex-A7 SoCs 32 bit RAM bus width may influence this result. This question must be further investigated in future work.

Groupings 2, 3, 4 and 5 are all based on sequential scanning rather than random memory access. This means that these four groupings offer some measure of sustained memory bandwidth. The general profile of all four groups is consistent with the results obtained by the STREAM benchmark, with the Cortex-A15 obtaining the best results by a significant margin, followed by the Cortex-A7 and then finally by the Cortex-A9.



**Figure 1.** *pmbw* Bandwidth Grouping Results

### 3.3. Discussion and Analysis

A clear correlation between age of SoC design and bandwidth efficiency (as a percentage) is observed, with the newest SoC, the Cortex-A15 performing the most effectively and the oldest SoC here, the Cortex-A9 performing the least effectively. Preliminary results which are to be presented at SAIP 2014 by Mitchell Cox [8] show that it is possible to obtain I/O connection rates between two Cortex-A9 based SoCs of approximately 300 MB/s. These results suggest that memory performance is not the primary source of throughput rate bottlenecks for relatively simple algorithms (i.e. where CPU performance is not the bottleneck), as this figure is approximately 5 times lower than the sustained memory bandwidth measured for the Cortex-A9. As I/O connection rates continue to improve, this low sustained memory bandwidth may present an obstacle to throughput rates. The Cortex-A9 tested here is, however, the oldest design of the three and the significantly improved memory bandwidth rates and efficiencies measured for the Cortex-A15 in particular means that memory bandwidth remains less likely to be the primary cause of throughput rate bottlenecks than I/O capacity for algorithms which are not highly processor intensive. These improvements are expected to continue as newer ARM-based SoCs are released, particularly with the soon-to-be released ARMv8 architecture 64 bit SoCs. The

potential of ARM-based SoCs for use in Data Stream Computing systems therefore remains strong. In addition to this, it is hoped that new Intel Atom hardware (i.e. development boards) will be procured in due course, allowing for further research and testing to be carried out.

### Conclusion

In conclusion, the paradigm or framework underpinning Data Stream Computing and its relevance to high-energy physics has briefly been discussed. The potential role of ARM-based SoCs in providing a solution to excess data production of large scale scientific experiments and the relevance of memory performance to this has also been discussed. The memory performance of three ARM-based SoCs has been evaluated and the implications of these results have been discussed. Finally, potential steps for the continuation of this research and development have been briefly outlined.

### References

- [1] Manyika J, Chui M, Brown B, Bughin B, Dobbs R, Roxburgh C and Hung Byers A 2011. *Big data: The next frontier for innovation, competition and productivity* (New York City, NY: McKinsey Global Institute)
- [2] Cox M, Reed R, Wrigley T, Harmsen G, and Mellado B 2014 *Performance Characterisation of ARM Cortex-A7, A9 and A15 System on Chips for Data Stream Computing* (submitted to Journal of Computational Science).
- [3] Aroca RV and Gonalves LMG 2012 Towards green data centers: A comparison of x86 and ARM architectures power efficiency, *Journal of Parallel and Distributed Computing* **72** 1770-80.
- [4] Dongarra, J., and Heroux, M. A. 2013 Toward a new metric for ranking high performance computing systems No. SAND2013-4744 312 (Albuquerque, NM: Sandia National Laboratories).
- [5] Ang, J A, Barrett B W, Wheeler, K B and Murphy R C 2010 *Introducing the graph 500* No. SAND2010-3263C (Albuquerque, NM: Sandia National Laboratories).
- [6] Bingmann, T. 2013 *pmbw Parallel Memory Bandwidth Benchmark/Measurement*. Retrieved from: <http://panthema.net/2013/pmbw/>
- [7] Larson, M G 2008 Analysis of variance *Circulation* **117.1**, 115-121
- [8] Cox, M. 2014 The development of a general purpose Processing Unit for the upgraded electronics of the ATLAS detector Tile Calorimeter, *SAIP 2014* (submitted).