Variability Windows for Predictable DDR Controllers, A Technical Report

MOHAMED HASSAN

1 INTRODUCTION

In this technical report, we detail the derivation of the variability window for the eight predictable memory controllers used in our RTSS paper [1]: AMC [2], PMC [3], RTMem [4], DCmc [5], ORP [6], MCMC [7], ROC [8], and ReOrder [9, 10]. As described in [1], we define the variability window by Definition 1.1.

Definition 1.1. **Variability Window** of a latency component, VW, is a measure of the possible variations in the value of this component and is computed as the percentage increase from the best (BCL) to the worst case latency (WCL) values of this component.

$$\text{VW} = \frac{WCL - BCL}{BCL} \times 100 \tag{1}$$

Accordingly, to compute VW for each of the eight DDRx controllers, we need to computer both the worst- (WCL) and best-case latencies (BCL).

2 WORST-CASE LATENCY

For the WCL, we use the generalized model provided in provided in [11]. In particular, we use Equations 2 and 3 from [11] for open- and close-page controllers (replicated by Equations 2 and 3 in this report), respectively. *HR* is the row hit ratio of the task and *REQr* is either the number of requestors in the same rank as the requestor under analysis (for controllers with rank support), or the total number of requestors in the system (for controllers without rank support).

$$Latency^{Req} = BasicAccess + Interference \cdot (REQr - 1) \tag{2}$$

$$\begin{aligned} Latency^{Req} = {} & (BasicAccess + RowAccess \cdot (1 - HR)) + \\ & (Interference + RowInter \cdot (1 - HR)) \cdot (REQr - 1) \end{aligned} \tag{3}$$

The terms *BasicAccess*, *RowAccess*, $Interference$, and $RowInter$ are controller dependent and are defined in Table 1 (a replication of Table 3 in [11]). In Table 1, *BI* determines the number of banks accessed by a request, *BC* determines the number of read (R) or write (W) commands generated for each bank, while *R* represents the number of ranks.

In [1], we derive the VW for a system with four processors. For controllers aimed at multi-rank DRAMs (MCMC, ROC, and ReOrder), we calculate the VW for a 4-rank memory system ($R = 4$ in Table 1). Accordingly, In Equations 2 and 3, as well as in Table 1: $REQr = 4$ for controllers that do not support multi ranks, while $REQr = 1$ for controllers supporting multi ranks. It is worth noting that PMC has a new version that supports multi-ranks [12]; however, this analysis only covers the version with no multi-rank support in [3]. For controllers that require knowledge about the hit ratio (DCmc, ORP, ROC, and ReOrder), we assume a hit ratio of $HR = 35\%$. Hit ratio is the percentage of requests accessing a row that is already existing in the row buffer. Computing the WCL for different number of ranks or different hit ratios is not the focus of this paper and is already studied in the the corresponding papers of these controllers as well as in the comparative study in [11]. Finally, we derive the analysis for a single memory access (i.e. $BI = 1$ and $BC = 1$). Based

Author's address: Mohamed Hassan.

Table 1. MC General Equation Components ($\mathcal{K}(cond)$ equals 1 if $cond$ is satisfied and 0 otherwise.)

	$RowInter$	$Interference$	$BasicAccess$	$RowAccess$
AMC	NA	$\big(15 \cdot \mathcal{K}(BI = 8) + 42\big) \cdot BC$	$\big(15 \cdot \mathcal{K}(BI = 8) + 42\big)$	NA
PMC RTMem	NA	$\mathcal{K}(BC = 1) \cdot \Big((15 \cdot \mathcal{K}(BI = 8) + 42)\Big) + \mathcal{K}(BC > 1) \cdot \Big((4 \cdot BC + 1) \cdot BI + 13 + 4 \cdot \mathcal{K}(BI = 8)\Big)$	$\mathcal{K}(BC = 1) \cdot \Big((15 \cdot \mathcal{K}(BI = 8) + 42)\Big) + \mathcal{K}(BC \neq 1) \cdot \Big((4 \cdot BC + 1) \cdot BI + 13 + 4 \cdot \mathcal{K}(BI = 8)\Big)$	NA
DCmc	0	$28 \cdot BC$	$13 \cdot BC$	18
ORP	7	$13 \cdot BC$	$19 \cdot BC + 6$	27
ReOrder	$7 + 3R$	$8R \cdot BC$	$(8R + 25) \cdot BC$	$33 + 3R$
ROC	$3 \cdot R + 6$	$\big(3 \cdot R + 12\big) \cdot BC$	$\big(3 \cdot R + 24\big) \cdot BC + 6$	$3 \cdot R + 27$
MCMC	NA	$Slot \cdot R \cdot BC$ Where $Slot = \begin{cases} 42/PE & if(REQr \leq 6) \wedge (R \leq 2) \\ 9 & if(R = 2) \wedge (REQr > 6) \\ 7 & Otherwise \end{cases}$	$Slot \cdot R \cdot BC + 22$	NA
FR-FCFS	0	$224 \cdot BC$	$24 \cdot BC$	18

Table 2. JEDEC Timing Constraints [13].

Parameter	Delay Description	Cycles
$tRCD$	A to R/W	10
$tCCD$	R to R or W to W (same rank)	4
tRL	R to start of data transfer	10
tRP	P to A	10
tWL	W to start of data transfer	9
$tRTW$	R to W	6
$tRTP$	R to P	5
$tWTR$	End of data transfer of W to R	5
tWR	End of data transfer of W to P	10
$tRAS$	A to P	24
tRC	A to A (same bank)	34
$tRRD$	A to A (diff bank in same rank)	4
$BL/2$	Data bus transfer	4 ($BL8$)*
$tRTRS$	Rank to rank switch	1

*BL is the burst length, which indicates the number of data beats to be transfered by one access.

on all these substitutions as well as the timing constraints for DDR3 shown in Table 2, we delineate the analytical WCL in Figure 1.

3 BEST-CASE LATENCY

We compute the BCL assuming the considered request does not suffer from any additional delays due to other requests. In other words, the BCL is due to timing constraints for only commands of the request under analysis. We have two cases, based on the controller type. For open-page controllers, in best case, a request will consist of a single R (or W) command. Therefore, the data will start transferring on the bus as soon as the corresponding tRL (or tWL) constraint is satisfied. From the JEDEC DDR DRAM standard (e.g. for DDR3 [13] as shown in Table 2) , $tWL \leq tRL$. Therefore, in best case the request is a W request. Figure 2 delineates this scenario. Accordingly, the BCL for open-page controllers is computed as

$$BCL^{Open} = tWL = 9\ cycles = 13.5ns.$$

On the other hand, for close-page controllers, the request has to start with an Activate (A) command followed by a write command W after $tRCD$ cycles. Figure 3 illustrates this scenario.

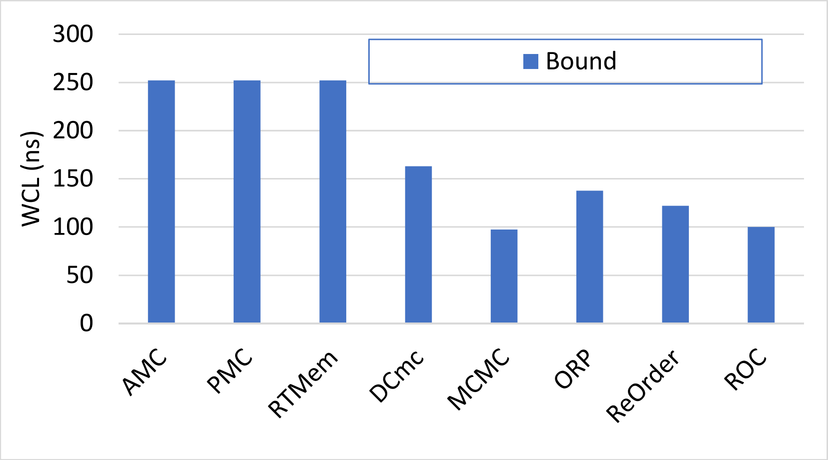

Fig. 1. Worst-case analytical latencies for the DDRx controllers for DDR3-1600 (1 cycle = 1.5ns).

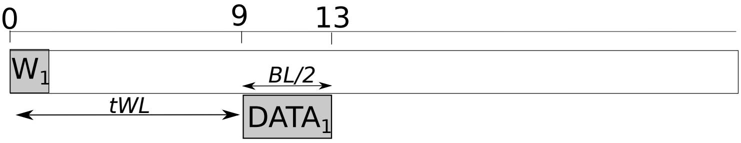

Fig. 2. Best-Case analytical latency for the DDRx controllers with open-page policy for DDR3-1600 (1 cycle = 1.5ns).

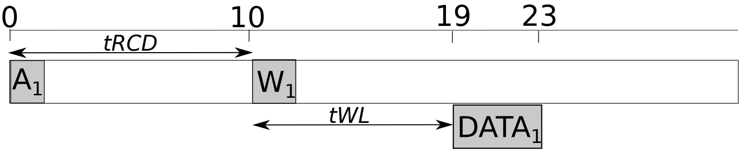

Fig. 3. Best-Case analytical latency for the DDRx controllers with close-page policy for DDR3-1600 (1 cycle = 1.5ns).

Hence, the BCL for close-page controllers is computed as

$$BCL^{Close} = tRCD + tWL = 19\ cycles = 28.5ns.$$

4 VARIABILITY WINDOW

Based on the derived WCL and BCL, the final step is to calculate the analytical VW using Definition 1.1. Figure 4 delineates the VW values for the considered DDRx controllers.

As Figure 4 illustrates, the variability window of these predicable controllers is huge. It exceeds 800% in 6 out of the 8 studied controllers. We observe that controllers with multi-rank support (MCMC, ROC, and ReOrder) provide less variability window. For instance, MCMC has the least variability window of 261%. This is because these controllers mitigate the interference among different PEs by partitioning banks among PEs as well as mitigates the bus switching delays by

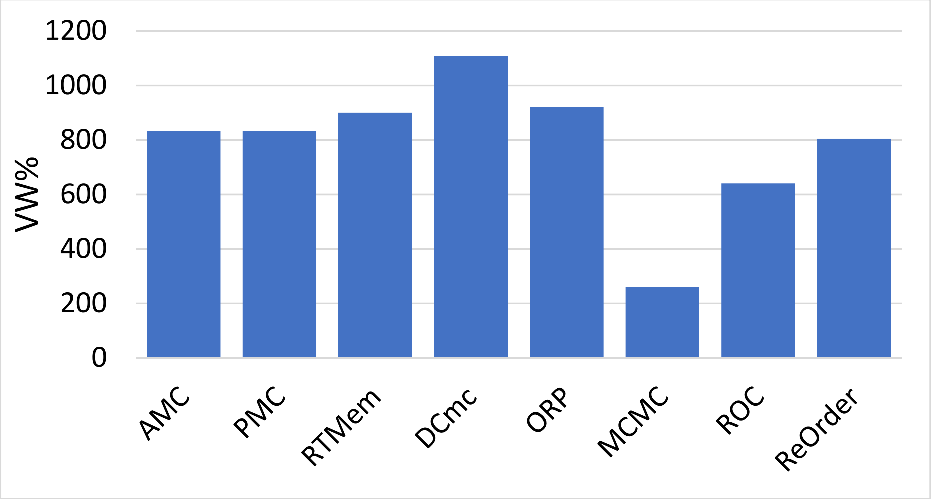

Fig. 4. Analytical variability window for different predictable memory controllers.

alternating between different ranks. However, the is still large and can be ill-suited for real-time systems with tight safety-critical timing requirements.

As aforementioned, this high variability is due to the physically inherent limitations of the DDRx memories that induce large timing constraints, which all controllers have to satisfy. As a result, we believe that exploring other types of off-chip memories that address these limitations is unavoidable towards providing more predictable memory performance with less variability and tighter bounds.

REFERENCES

[1] Mohamed Hassan. On the off-chip memory latency of real-time systems: Is ddr dram really the best option? In *IEEE Real-Time Systems Symposium (RTSS)*, 2018.

[2] Marco Paolieri, Eduardo Quiñones, Fransisco J. Cazorla, and Mateo Valero. An analyzable memory controller for hard real-time CMPs. *Embedded System Letters (ESL)*, 1:86–90, 2009.

[3] Mohamed Hassan, Hiren Patel, and Rodolfo Pellizzoni. A Framework for Scheduling DRAM Memory Accesses for Multi-Core Mixed-time Critical Systems. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 307–316, 2015.

[4] Yonghui Li, Benny Akesson, and Kees Goossens. Dynamic command scheduling for real-time memory controllers. In *Euromicro Conference on Real-Time Systems (ECRTS)*, pages 3–14, 2014.

[5] Javier Jalle, Eduardo Quinones, Jaume Abella, Luca Fossati, Marco Zulianello, and Francisco J Cazorla. A dual-criticality memory controller (DCmc): Proposal and evaluation of a space case study. In *IEEE Real-Time Systems Symposium (RTSS)*, 2014.

[6] Zheng Pei Wu, Yogen Krish, and Rodolfo Pellizzoni. Worst case analysis of DRAM latency in multi-requestor systems. In *Real-Time Systems Symposium (RTSS)*, pages 372–383, 2013.

[7] Leonardo Ecco, Sebastian Tobuschat, Selma Saidi, and Rolf Ernst. A mixed critical memory controller using bank privatization and fixed priority scheduling. In *IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2014.

[8] Yogen Krishnapillai, Zheng Pei Wu, and Rodolfo Pellizoni. ROC: A rank-switching, open-row DRAM controller for time-predictable systems. In *Euromicro Conference on Real-Time Systems (ECRTS)*, 2014.

[9] Leonardo Ecco and Rolf Ernst. Improved DRAM timing bounds for teal-time DRAM controllers with read/write bundling. In *Real-Time Systems Symposium*, pages 53–64, 2015.

[10] Leonardo Ecco, Adam Kostrzewa, and Rolf Ernst. Minimizing DRAM rank switching overhead for improved timing bounds and performance. In *Euromicro Conference on Real-Time Systems (ECRTS)*, 2016.

[11] Danlu Guo, Mohamed Hassan, Rodolfo Pellizzoni, and Hiren Patel. A comparative study of predictable DRAM controllers. *ACM Transactions on Embedded Computing Systems (TECS)*, 2018.

[12] Mohamed Hassan, Hiren Patel, and Rodolfo Pellizzoni. PMC: A requirement-aware DRAM controller for multi-core mixed criticality systems. In *ACM Transactions on Embedded Computing Systems (TECS)*, 2016.

[13] JEDEC DDR3 SDRAM specifications jesd79-3d. http://www.jedec.org/standards-documents/docs/jesd-79-3d. Accessed: 2018-09-12.