Research Centre in Real-Time Computing Systems
FCT Research Unit 608

Technische Universiteit Eindhoven University of Technology

An Optimal Multi-Channel Memory Controller for Real-Time Systems

Manil Dev Gomony, Benny Akesson, and Kees Goossens

Problem statement

→ ***Devise a multi-channel memory controller architecture and a method for mapping of memory clients to the memory channels for minimum resource utilization while meeting the real-time requirements of the clients.***

Multi-channel memories

JEDEC Wide IO DRAM standard:

→ *Introduced in December 2011.*

→ *Four independent DRAM channels with dedicated data bus and control signals.*

→ *Data bus of each channel is 128-bit wide.*

→ *Every channel has 4 banks.*

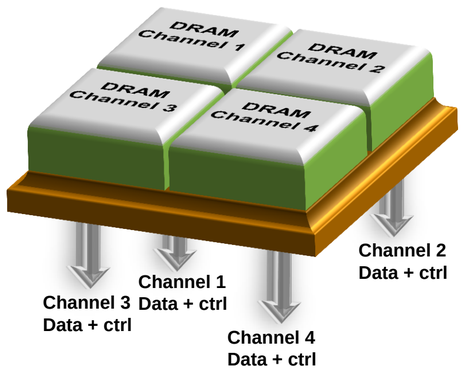


Figure 1: Four-channel Wide IO DRAM

→ *Only Wide IO based DRAMs can meet the power and bandwidth demands of future mobile devices [1].*

→ *Provides up to 3.1 GB/s of guaranteed bandwidth per memory channel.*

→ *Reduces application execution time by 25% and power consumption by 67% compared to LPDDR memories.*

Multi-channel memory controller

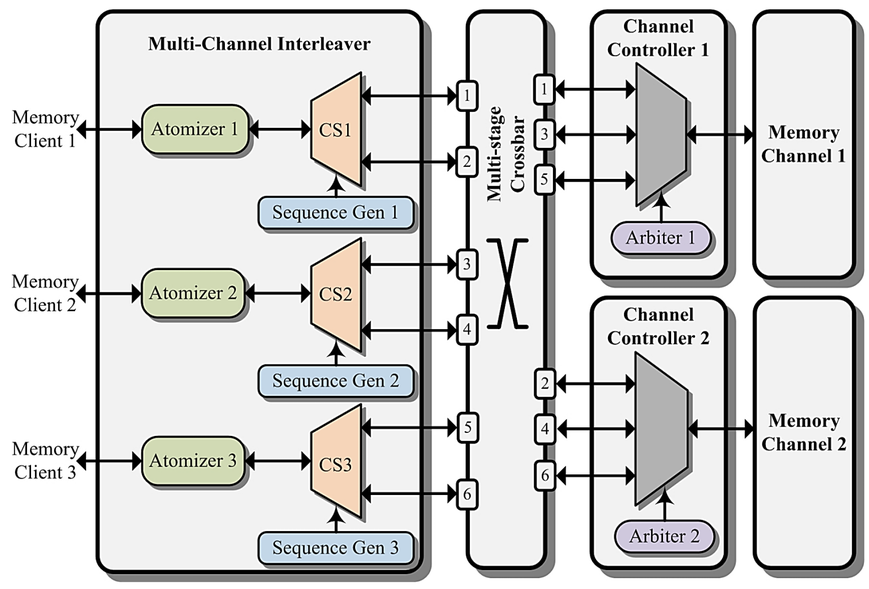


Figure 2: Real-Time multi-channel memory controller architecture [2]

→ *The incoming memory requests are split into a number of smaller sized service units by the Atomizer.*

→ *The service units are routed to the different memory channels by the Channel Selector (CS) according to the configuration programmed in the Sequence Generator.*

→ *The CS consists of separate request and response paths as shown in Figure 3. The address translator performs the logical-to-physical address conversion (see below).*

→ *The (read) responses from different channels are buffered in the CS since they may arrive out-of-order.*

→ *We use existing real-time memory controllers as Channel Controllers which bound the execution time of a memory transaction by fixing the memory access parameters, such as burst length, number of read/write commands etc., at design-time.*

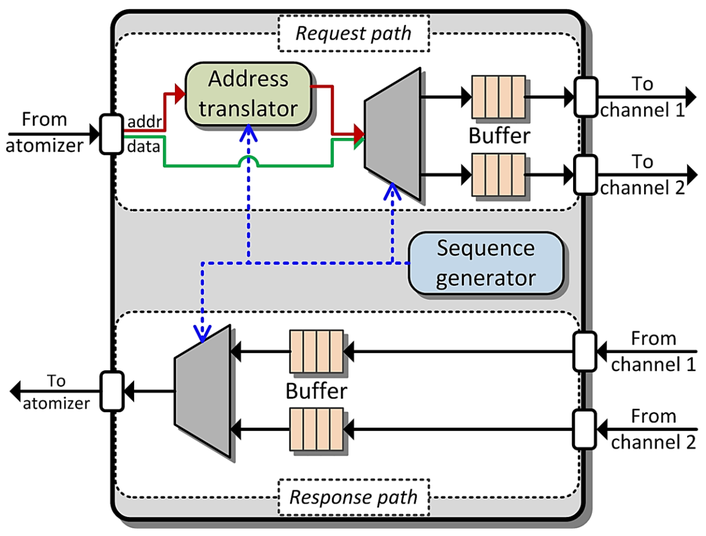


Figure 3: Channel Selector architecture showing both request and response paths

Logical-to-physical address translation

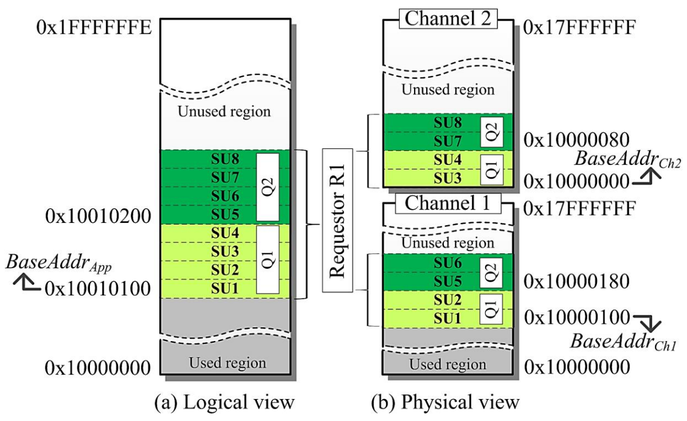


Figure 4: Example memory map showing requestor R1 allocated to two memory channels, with every request Q1 and Q2 interleaved across the two channels.

→ *Service Units (SU) of a memory request can end up in different physical addresses in each channel*

→ *The Logical view of the memory space must be continuous to avoid explicit data partitioning and data movement while writing the application program*

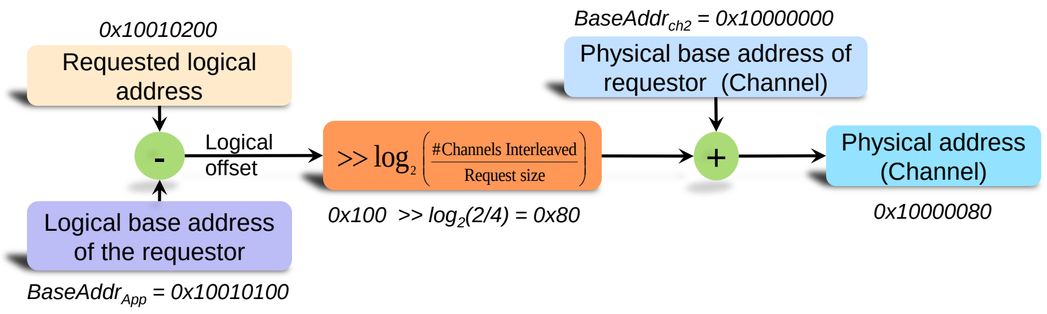
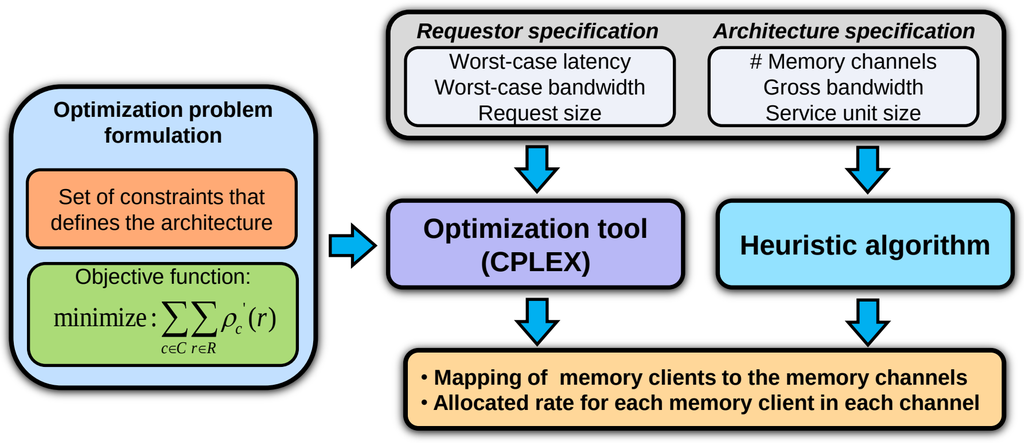


Figure 5: Logical-to-physical address computation with example

Mapping of memory clients to memory channels

→ ***Optimization goal:*** *Map memory clients to the memory channels for minimum bandwidth (rate) utilization while meeting the real-time requirements of the clients.*



Number of channels	Number of memory clients	CPLEX	Heuristic
4	25	8.9 secs	< 1 sec
	50	2.2 mins	
	100	13.4 mins	
8	25	59.6 mins	< 1 sec
	50	6 hrs	
	100	2 days	
16	25	> 3 days	< 1 sec
	50		
	100		

→ ***Heuristic algorithm is the only scalable solution for future system requirements!***

References

[1] M.D.Gomony et al., "DRAM Selection and Configuration for Real-Time Mobile Systems." in Proc. DATE, 2012.

[2] M.D.Gomony et al., "Architecture and Optimal Configuration of aReal-Time Multi-Channel Memory Controller." in Proc. DATE, 2013.

Manil Dev Gomony
m.d.gomony@tue.nl

Acknowledgements

This work was partially funded by projects EU FP7 288008T-CREST and 288248 Flextiles, Catrene CA104 COBRA,ARTEMIS 100202 RECOMP, PT FCT, and NL STW 10346NEST.