An Energy Efficient DRAM Subsystem for 3D integrated SoCs

Christian Weis†, Igor Loi⋆, Luca Benini⋆ and Norbert Wehn†
†Microelectronic Systems Design Research Group, TU Kaiserslautern, Kaiserslautern, Germany
⋆DEIS, University of Bologna, Bologna, Italy
{weis, wehn}@eit.uni-kl.de {igor.loi, luca.benini}@unibo.it

***Abstract*—Energy efficiency is the key driver for the design optimization of System-on-Chips for mobile terminals (smartphones and tablets). 3D integration of heterogeneous dies based on TSV (through silicon via) technology enables stacking of multiple memory or logic layers and has the advantage of higher bandwidth at lower energy consumption for the memory interface. In this work we propose a highly energy efficient DRAM subsystem for next-generation 3D integrated SoCs, which will consist of a SDR/DDR 3D-DRAM controller and an attached 3D-DRAM cube with a fine-grained access and a very flexible (WIDE-IO) interface. We implemented a synthesizable model of the SDR/DDR 3D-DRAM channel controller and a functional model of the 3D-stacked DRAM which embeds an accurate power estimation engine. We investigated different DRAM families (WIDE IO DDR/SDR, LPDDR and LPDDR2) and densities that range from 256Mb to 4Gb per channel. The implementation results of the proposed 3D-DRAM subsystem show that energy optimized accesses to the 3D-DRAM enable an overall average of 37% power savings as compared to standard accesses. To the best of our knowledge this is the first design of a 3D-DRAM channel controller and 3D-DRAM model featuring co-optimization of memory and controller architecture.**

I. Introduction

To overcome the pin-limited performance growth [1], the power vs. bandwidth dilemma and the memory wall *3D integration* and *3D-stacked DRAM* have been proposed as a very promising solution. 3D-stacked DRAMs reduce the distance between CPU or GPU and external DRAM from centimeters to micrometers and improve the bandwidth and access latencies - but more importantly, they provide a major boost in energy efficiency in comparison to standard DRAM devices, such as DDR2 or DDR3 [2]. The pairing of high bandwidth communication with the lower power consumption of 3D integrated DRAM is an ideal fit for mobile terminals.

In the last years 3D integration of ICs, especially of DRAMs, received tremendous attention [2]–[8]. We recognized that with the WIDE IO DRAM (4x 128 IOs) standard a much higher bandwidth (at 200MHz: 12.8 GB/s) is available for mobile SoCs. However if the full bandwidth is not needed during a read or write, a lot of data is wasted and also the energy for the transfer.

We focus in this paper on the energy optimization of the 3D-DRAM subsystem for future terminals. We propose a flexible bandwidth and burst length adaption for the 3D-DRAM and the controller. With this we are able to handle a large range of access sizes from fine-grained (32b) to coarse-grained (1Kb). The main contributions of this work are

1) the co-optimization of controller and 3D-DRAM,
2) the fine-grained access to the 3D-DRAM which leads to power savings and an energy proportional access,
3) the implementation of the 3D-DRAM controller which covers all features needed for such flexible interface.

978-3-9810801-8-6/DATE12/©2012 EDAA

II. Subsystem Architecture

This section gives an overview about the complete subsystem, shows details of the controller and 3D-DRAM, as well as describes the used 3D-DRAM configurations. The functional representation of the 3D-DRAM subsystem is shown in Figure 1, which consists of a 3D-DRAM memory controller and 3D-DRAM channels. The controller is divided into front-end (FE) and back-end (BE). The front-end is responsible for request handling, including FIFO-buffers for each request channel (RC), the arbitration and scheduling. The back-end contains the channel controllers (CC) for each 3D-DRAM channel and is responsible for the DRAM command encoding and communication. Each CC is attached to the 3D-DRAM channels via a very flexible interface which provides up to 128 data IOs to each 3D-DRAM channel. The aim of this flexible interface is to adjust the bandwidth to/from the 3D-DRAM depending on the incoming request (RC 0 to 2). We have identified three typical request and bus sizes for integration of this subsystem into a heterogeneous SoC:

- **32-bit**, AHB/AXI (ARM processor, RC **0**)
- **64-bit**, Graphics/Imaging (mobile GPU, RC **1**)
- **128-bit**, Video Encoder (HD processing core, RC **2**)

A. 3D-DRAM channel controller for SDR/DDR

The channel controller is the back-end part of the memory controller, and manages incoming transactions from the front-end side and passes them to a specific memory channel. This IP has been designed to work seamlessly with the 3D-DRAM flexible interface and get the maximum benefits in terms of power efficiency and performance. This block is highly configurable and has the ability to work with several memory types (SDR/DDR/LPDDR) and different flavors (number of banks, IO width, etc). Additionally some features can be reconfigured at run-time writing into dedicated registers. The internal architecture of the channel controller is depicted in Figure 2 and consists of a first buffering stage, a DRAM

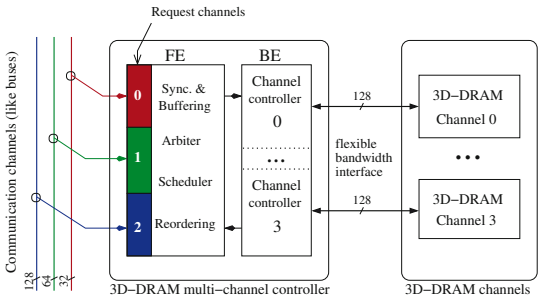

Fig. 1. 3D-DRAM subsystem - functional overview (including front-end (FE) and back-end (BE) of the controller, the request channels (RC) and the channel controllers (CC))

command encoder, a lite physical interface, a reconfigurable interface block and IO circuits for the DRAM ports.

The buffering stages are needed to provide resources to temporarily store incoming requests, and to synchronize data transfer when crossing a clock domain boundary (FE to BE). The front-end usually is clocked at high speed to allow complex operations like scheduling and reordering, while the back-end is locked to the memory frequency. The DRAM command encoder decodes the incoming requests (post buffering) and generates the right command sequences to precharge a bank, activate a row, perform a read or write, refresh periodically the memory, provide the startup configuration and safe power up. These tasks are managed in several control units and scheduled by a single master control block, which synchronizes these control units. Each bank is managed by a dedicated control unit, which tracks all the activity related to a specific bank (e.g. active row) in order to maximize the performance and minimize the power consumption. The *flexible bandwidth adaptation* is managed in each bank control unit and depending on the type of access, it provides for the memory the correct signaling needed to perform flexible data accesses (32/64-bit).

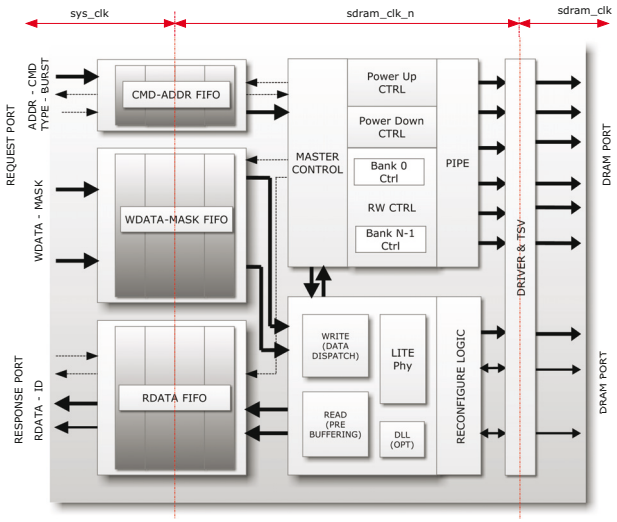

Fig. 2. 3D-DRAM channel controller (CC)

Strobe and data strobe are managed in the lite physical interface, which latches the incoming data from the memory in case of load, or preparing and dispatching write data in case of store. In case of SDR, the DRAM clock can be used to safely latch read data, or delivery write date. In case of DDR, a DLL is needed to create a 2x faster DRAM clock to safely latch and dispatch data. The reconfiguration logic block is used in 32/64-bit access modes, and the aim of this module is to selectively mask not driven data lines, and compact the data in the prefetch buffer which accommodates 128-bit words. Finally the last stage is composed by a pipe, to filter SDRAM signals from any kind of glitches, and providing more timing budget for inter-die traversal. The last stage is composed by dedicated IO circuits to drive the DRAM ports through a stack of TSV and micro-bumps.

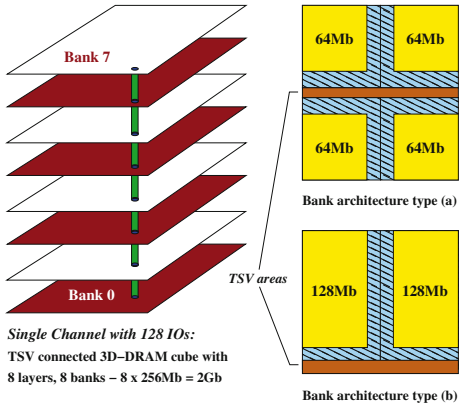

Fig. 3. Architectures of a 2Gb 3D-DRAM (single channel)

B. 3D-DRAM architecture and flexible bandwidth interface

The investigated 3D-DRAMs are closely aligned to the new Wide IO DRAM JEDEC standard. Figure 3 depicts the used two different *true* 3D-DRAM architectures. As in [9] explored an optimized 3D-DRAM with 8 banks consists of 8 layers (tiers) which corresponds to a layer per bank organization. For multiple banks per layer the optimal architecture must be revised and details are not shown here because of space reason. The two architectures differ in the tile size used for a 3D-DRAM macro block to compose a bank. For bank architecture type (a) the tile size is 64Mb and for type (b) 128Mb. Option (b) has a higher area efficiency and therefore lower cost with the impact of lower performance. In Figure 3 the yellow areas with tile size numbers show the DRAM cell arrays, the light blue areas with stripes are occupied by column, row and control circuits as well as all other peripheral circuits and the brown areas, as indicated, are reserved for the vertical TSV connections.

Our 3D-DRAM architecture is characterized by an implementation of a *flexible bandwidth* interface which enables internal organization switching for the 3D-DRAM. This is shown in Figure 4. The 3D-DRAM is able to operate in three different configurations:

- Native mode **x128**: All column select lines (16 CSLs) and all wordlines (2 WLs) are activated.
- Half bank mode **x64**: Only 8 CSLs and 1 WL are activated.
- Quarter bank mode **x32**: Only 4 CSLs and 1 WL are activated.

The multiplexer circuits for switching the data chunks, as indicated in Figure 4, are placed only once on the logic die (channel controller). This reduces the area overhead for an 8-layer 3D-DRAM cube and improves performance, power and access latency.

To enable or disable these operating modes new Mode Register (MR) entries have to be defined. We have integrated them into the Extended Mode Register (EMR) as this usually done for new entries, see Figure 5. Additionally to the EMR settings new signals have to be defined for the 3D-DRAM interface in order to switch the organization *on-the-fly*. The names of these signals were already introduce in Figure 4 and 5. The functional details of those are:

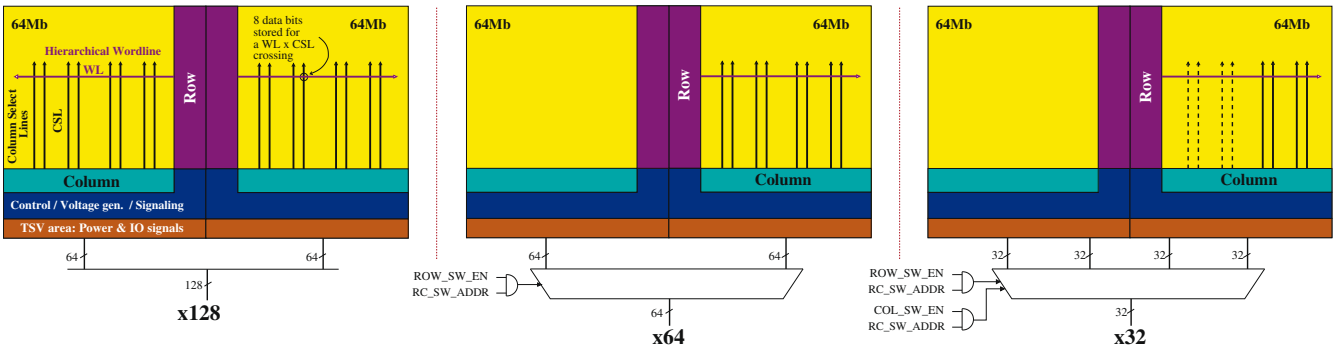

Fig. 4. Flexible 3D-DRAM organization/bandwidth switching of a 128Mb bank in a 8 bank 1Gb 3D-DRAM, operating in x128, x64 or x32 mode

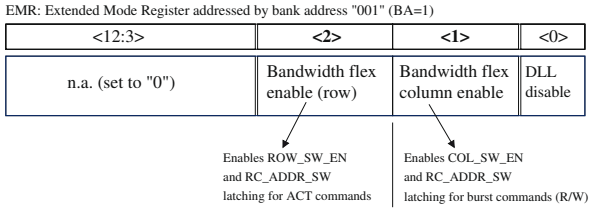

Fig. 5. EMR settings for the flexible bandwidth interface of 3D-DRAM

- ROW_SW_EN - enables during a ACT command the organization switch.
- COL_SW_EN - enables during a RD/WR command the organization switch, depending on the value of ROW_SW_EN, which is stored for the each bank.
- RC_SW_ADDR - address which decides on the selection of the data chunk.

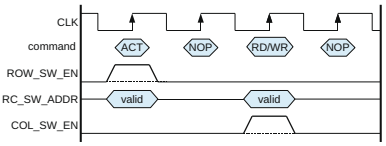

Fig. 6. Timing diagram

Figure 6 shows the input timing behavior of the new interface signals related to a ACT and RD/WR command. If the new interface signals are enabled via the EMR and ROW_SW_EN or COL_SW_EN are asserted during a ACT or WR/RD command respectively, then a valid RC_SW_ADDR must be provided.

C. WIDE IO DRAM Configurations

Table I shows the summary of the generated 3D-DRAM SDR and DDR configurations. Typical performance, area and technology data are given here. In order to be JEDEC conform the supply voltage is set to 1.2V for all configurations. The footprint for the single channel can be calculated by $A_{footprint} = Area/\#\ of\ lay$. In contrast to [5] we used a TSV diameter value of 8 μm and 16 μm pitch. This diameter is very similar to the ones Samsung uses in [10] d=7.5 μm but with a pitch of 50 μm given by JEDEC. Our TSV capacitance evaluates to 94 fF and TSV resistance to 23 $m\Omega$ by using copper as filling material. Overall ten different 3D-DRAM configurations were created to strengthen our experiments.

III. Simulation and Traffic Generation

Following simulation setup and infrastructure was implemented to verify the functionality of the 3D-DRAM subsystem

TABLE I
3D-DRAM x128 single channel configurations

Dens Mb	Arch type	# of lay	# of banks	Tech nm	Cell size	Area mm^2	Freq - IF MHz
*256	n.a.	1	4	58	$6F^2$	16	200 - SDR
512	n.a.	2	4	58	$6F^2$	26	200 - SDR
1024	(b)	8	8	46	$6F^2$	35	300 - SDR
2048	(b)	8	8	46	$6F^2$	60	167 - SDR
4096	(a)	8	8	45	$4F^2$	97	200 - SDR
256	n.a.	1	4	58	$6F^2$	22	200 - DDR
512	n.a.	2	4	58	$6F^2$	32	200 - DDR
1024	(a)	8	8	46	$6F^2$	44	300 - DDR
2048	(a)	8	8	46	$6F^2$	69	300 - DDR
4096	(a)	8	8	45	$4F^2$	98	200 - DDR

* *Density emulates the published Samsung 1Gb WIDE IO chip [10].*

and to run our experiments. Three different traffic generators emulate the workloads produced by a typical SoC platform as it will be implemented in future mobile terminals:

- Traffic A: 32-bit - Cache misses of an ARM - 0.1 GB/s
- Traffic B: 64-bit - DMA accesses in a SoC - 0.8 GB/s
- Traffic C: 128-bit - HD Video DMA accesses - 1.5 GB/s

We used 2x the traffic generator A - to simulate a Dual-Core environment. Additionally we enabled 1x traffic generator B and C. B emulates an imaging unit with DMA accesses and C an HD video core. Altogether we created a bandwidth (BW) request of 2.5 GB/s to the 3D-DRAMs. For the LPDDRx32-333 and LPDDR2x32-667 based on 2Gb Micron data sheets we reduced the bandwidth to 750 MB/s. These generators can be tuned to simulate different workload scenarios, see Table II. The page hit ratio (PHR) describes the relationship between

TABLE II
Aggregated Workloads

Name	Page Hit Ratio [%]	Read [%]	Write [%]	BW [GB/s]
IDLE	n.a.	0	0	0
PHR0, PHR50, PHR100	0, 50, 100	60	40	2.5

bursts for which a page (activate a row) must be opened and bursts which are directed to an already opened page. If the PHR is 0% then for each burst an activate command to open a new page is required. If the PHR is 100% then all bursts are issued to an already opened page.

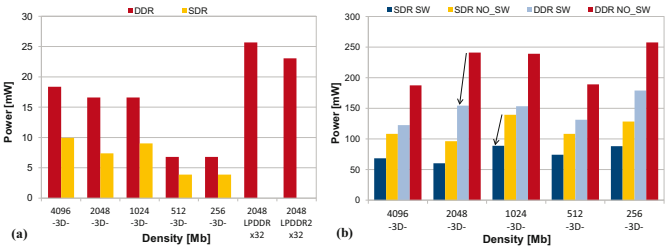

Fig. 7. Power characterization of the DRAMs, (a) IDLE, (b) PHR=50%

IV. Experimental Results

In all experiments we present here we used a single channel of the 3D-DRAM cube connected to the flexible bandwidth and burst length adaption interface of the controller. We considered in our analysis only a single channel to put emphasis on the interface and the power savings per channel. Thus the results can be scaled when multiple channels or slices [2] are used. A single channel allows us also a fair and valid comparison to LPDDR/LPDDR2 devices. However, we had to scale down the applied bandwidth to 750 MB/s for LPDDRx32-333 or LPDDR2x32-667 devices as they support only peak bandwidths up to 1.33 or 2.66 GB/s respectively.

A. DRAM Power characterization

First we characterized the ten 3D-DRAM configurations and LPDDR/LPDDR2 devices by using the workload IDLE. During IDLE only AREF (Refresh) commands are sent to the DRAM. Figure 7 (a) shows the differences in Idle power for DDR and SDR mode. In DDR mode a DLL (Delay-Locked-Loop) is additionally enabled and it contributes significantly to the power consumption. We see also the effect that the leakage current is increased for higher densities.

For the workload with a PHR of 50%, which is equivalent to 50% page miss, each second data access to the 3D-DRAM opens and closes a *new* row. The results for this application-relevant mode are maximal 39% power savings between bandwidth switching enabled (**SW**) and disabled (**NO_SW**) for the 2048Mb density in SDR mode. The average power savings of this mode for all densities and interfaces (SDR, DDR) are **34.1%**, see also Figure 7 (b). The average savings of PHR=0% and PHR=100% are **34.9%** and **43.2%** respectively. We ommitted the plots of those because they are very similar. Finally Figure 8 shows the advantage of 3D-DRAMs over LPDDR/LPDDR2 x32 devices.

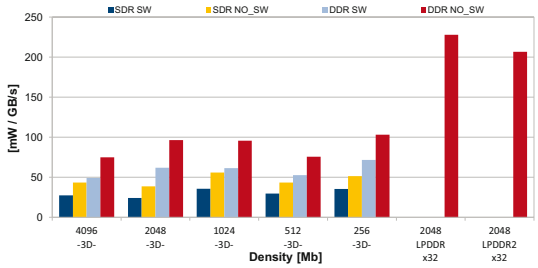

Fig. 8. Comparison to LPDDR/LPDDR2 with PHR=50%

B. Synthesis Results of the SDR/DDR channel controller

In this section, we discuss the experimental results for the SDR/DDR channel controller in terms of power and area. To get these results, we synthesized the controller with the ST65nm technology library (Low Power). The front-end flow (Multi V_{TH}) has been performed with Synopsys Design Compiler in topographical mode, while the back-end with Cadence SoC Encounter. To run the synthesis flow, we fixed the frequency on both DRAM and front-end side choosing 500MHz and 333MHz respectively. The power cost is dominated by clock distribution network (40%) and sequential elements (45% for buffering), while the combinational power impact is quite low (9%). The total power consumption (post-P&R) for this test case is 78mW, including the contribution of the IO drivers (6%). The total area cost is 280K μm^2 and is dominated by the read and write FIFOs (72%) and Lite PHY (16%). Since these results are dominated by buffering resources (FIFOs), the impact of power and area can be reduced by decreasing the depth of these storage elements.

V. Conclusion

In this paper we presented a new architecture for a highly energy efficient 3D-DRAM subsystem for 3D integrated SoCs. We investigated different DRAM families. 3D-DRAM densities from 256Mb to 4096Mb and also 2Gb LPDDR/LPDDR2 devices were characterized by four application-relevant workloads. We designed a 3D-DRAM channel controller which fits perfectly to the flexible bandwidth and burst length adaption interface of the 3D-DRAMs. By using this flexible interface the total 3D-DRAM subsystem consumes in the fastest configuration for 3D-DRAM (300MHz, 2Gb DDR) and controller (500MHz) less than *240 mW*, for a workload with PHR = 50%. With very low power settings of 3D-DRAM (167MHz, 2G SDR) and controller the subsystem power is decreased to less than *140 mW*. The experimental results show an overall average of **37%** power savings for 3D-DRAMs by enabling the bandwidth and burst length flexibility.

Acknowledgments

This work was partly supported by FP7 PRO3D GA n. 248776.

References

[1] P. Stanley-Marbell et al. , “Pinned to the walls – Impact of packaging and application properties on the memory and power walls,” in *Proc. ISLPED*, 2011, pp. 51–56.

[2] Micron Techn., Inc., “Hybrid Memory Cube: Breakthrough DRAM Performance with a Fundamentally Re-Architected DRAM Subsystem,” Hot Chips 23, Palo Alto, California, www.micron.com, Aug. 2011.

[3] D. Dutoit and A. Jerraya. (2010, July) 3D Integration Opportunities for Memory Interconnect in Mobile Computing Architectures. Issue 34. CEA. pp. 38-45. [Online]. Available: http://www.future-fab.com/

[4] R. Anigundi et al., “Architecture design exploration of three-dimensional (3D) integrated DRAM,” in *Proc. ISQED*, 2009, pp. 86–90.

[5] M. Facchini et al., “System-level power/performance evaluation of 3D stacked DRAMs for mobile applications,” in *Proc. DATE*, 2009.

[6] D. H. Woo et al., “An optimized 3D-stacked memory architecture by exploiting excessive, high-density TSV bandwidth,” in *Proc. IEEE 16th Int High Performance Computer Architecture Symp*, 2010, pp. 1–12.

[7] I. Loi et al., “An efficient distributed memory interface for many-core platform with 3D stacked DRAM,” in *Proc. DATE*, 2010, pp. 99–104.

[8] G. H. Loh, “3D-Stacked Memory Architectures for Multi-core Processors,” in *Proc. ISCA*, 2008, pp. 453–464.

[9] C. Weis, N. Wehn, L. Igor, and L. Benini, “Design Space Exploration for 3D-stacked DRAMs,” in *Proc. DATE*, 2011, pp. 1–6.

[10] J.-S. Kim et al., “A 1.2V 12.8GB/s 2Gb mobile Wide-I/O DRAM with 4x128 I/Os using TSV-based stacking,” in *ISSCC*, 2011, pp. 496–498.