25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Cheon Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1], Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1], Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1], Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1], Myeong Jun Song[1], Ahn Choi[1], Daeho Kim[1], SooYoung Kim[1], Eun-Bong Kim[1], David Wang[2], Shinhaeng Kang[1], Yuhwan Ro[3], Seungwoo Seo[3], JoonHo Song[3], Jaeyoun Youn[1], Kyomin Sohn[1], Nam Sung Kim[1]

[1]Samsung Electronics, Hwaseong, Korea
[2]Samsung Electronics, San Jose, CA
[3]Samsung Electronics, Suwon, Korea

2021 IEEE International Solid- State Circuits Conference (ISSCC) | 978-1-7281-9549-0/20/$31.00 ©2021 IEEE | DOI: 10.1109/ISSCC42613.2021.9365862

In recent years, artificial intelligence (AI) technology has proliferated rapidly and widely into application areas such as speech recognition, health care, and autonomous driving. To increase the capabilities of AI more powerful systems are needed to process a larger amount of data. This requirement has made domain-specific accelerators, such as GPUs and TPUs, popular; as they can provide orders of magnitude higher performance than state-of-the-art CPUs. However, these accelerators can only operate at their peak performance when they get the necessary data from memory as quickly as it is processed: requiring off-chip memory with a high bandwidth and a large capacity [1]. HBM has thus far met the bandwidth and capacity requirement [2-6], but recent AI technologies such as recurrent neural networks require an even higher bandwidth than HBM [7-8]. While a further increase in off-chip bandwidth can be accomplished by various techniques, it is often limited by power constraints at the chip or system level [9]. Hence, it is essential to decrease demand for off-chip bandwidth with unconventional architectures: such as processing-in-memory. In this paper, we present function-In-memory DRAM (FIMDRAM) that integrates a 16-wide single-instruction multiple-data engine within the memory banks and that exploits bank-level parallelism to provide 4× higher processing bandwidth than an off-chip memory solution. Second, we show techniques that do not require any modification to conventional memory controllers and their command protocols, which make FIMDRAM more practical for quick industry adoption. Finally, we conclude this paper with circuit- and system-level evaluations of our fabricated FIMDRAM.

Figure 25.4.1 shows the architecture of a conventional HBM2 and FIMDRAM. To maintain the same physical dimension as HBM2 and preserve mechanical compatibility with the existing 2.5D system-in-package, half of the cell array in each bank was removed and replaced with the programmable computing unit (PCU). Additionally, two banks share one PCU, and there are 8 PCUs per pseudo-channel: the location and the number of PCUs are optimized by considering the area overhead of a PCU and the number of simultaneous operation bank is considered by the reduction of peak noise. By locating the PCU adjacent to the cell array, the FIMDRAM is able to utilize bank-level parallelism; the number of simultaneously internal operations is maximized, and the overhead due to the data bus connected between the cell array and PCU is minimized. To support concurrent multi-bank operations a special FIM mode for FIMDRAM is added. During FIM mode, all PCUs are turned on simultaneously and multiple banks execute the same command from the host, while normal mode supports basic operation of the HBM2.

To control FIM operations without modification of the memory controller, an internal FIM controller was designed as shown in Fig. 25.4.2. The internal FIM controller consists of a mode generator to manage FIM mode entry and exit, and a clock divider to enable direct control the PCUs. FIM mode entry can be initiated by a row activate (ACT) command, and exit can be performed by ACT and precharge (PRE) commands. When the host wants to change the operation mode, it can do so by conventional row commands with a specific address: including setting the 13th row-address (RA) bit high. The most significant bit is used to switch in and out of FIM mode: this is to ensure safe operation by applying a high address bit to the memory region where the cell array was removed in FIMDRAM. Additionally, read (RD) and write (WR) commands, and the 13 RA are used to control the three types of data movement between cell array and PCU during FIM mode, as shown in Fig. 25.4.3. In addition, a multi-bank operation can be directed according to the bank address (BA): ACT/PRE/RD/WR commands can simultaneously control all even banks or all odd banks depending on BA0. Concurrent operations on multiple banks increases performance during FIM mode of operation without DRAM read/write latency, because the data bus for normal operations is same as the conventional one, and a local data bus for FIM operations is isolated to prevent any change to normal operations. Additionally, a data bus control scheme is adopted to reduce power consumption: for data movement between the cell array and PCUs the local data bus is active and the global one is not, since there is no need to move data to the external host. Thus, power impact is reduced due to processing operations in DRAM.

Figure 25.4.4 shows the PCU's block diagram and its instruction set. The PCU is divided into a register group, an execution unit, and an interface unit. To support universal and programmable FIM operations, the register group consists of a command-register file (CRF) for instruction memory, a general-purpose register file (GRF) for weight and accumulation, and a scalar register file (SRF) to store constants for MAC operations. Additionally, the PCU contains execution units, a decoding unit for parsing instructions needed to perform operations, and interface units to control data flow. The PCU is controlled by conventional memory commands from the host via the newly implemented control paths to enable the in-DRAM computations. The input signals to the PCU are divided into two groups. (1) Conventional signals for DRAM (internal read/write master signal, addresses, and DQs), and (2) generated signals to control PCUs: such as mode control signal and a generated clock for FIM operations. These signals control the instruction sequences for floating point matrix-vector calculations in the register group. The execution unit in the PCU uses a variable four-stage pipeline operation; depending on the decoded source operands, the stored data from the register files or cell data from the bank array, are selected and used as inputs for the execution unit by the decoded opcode. Additionally, the execution unit has a one stage pipeline for the JUMP instruction, which supports looping. Computed results from the execution units are directed by the destination operand and stored in the GRF. The updated GRF data can be moved into a target bank with a MOVE instruction.

Figure 25.4.5 shows FIMDRAM operation flow in normal and FIM mode; it is set to normal mode after initialization. During normal mode, the host can access each bank as a regular DRAM; it is used to set up input vectors for FIM mode. In order to operate PCUs, the host must send a command to change to FIM mode. After which, even or odd banks receive commands as a group and the PCU is ready to calculate in FIM mode. After that, the host issues a programming sequence and sends the weights for matrix-vector calculation via the DQ interface. The PCU can save 32 instructions in the CRF, and the instructions are sequentially read by CRF program counter. Upon completion of the computation phase, the results of the register files are transferred from PCU to memory cell arrays by multi-bank operations. To read the results from each bank, the host must switch back to normal mode, and then the stored data in each bank can be read.

The FIMDRAM was fabricated using a 20nm DRAM process. Figure 25.4.6 shows a key feature summary and the measurement results. These show that the FIMDRAM achieves 2.4Gbps/pin operation without increasing power consumption, and that the PCU can operate at 300MHz. In addition, an FPGA-based platform and an emulation environment based on an application model was developed to confirm system performance improvements by FIMDRAM. The evaluation results show a 2.1× improvement in performance with a 71% system energy reduction compared to a typical GPU system using HBM2. The chip micrograph is shown in Fig. 25.4.7.

References:

[1] N. Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit," *ISCA*, pp. 1-12, June 2017.
[2] D. Lee et al., "A 1.2V 8Gb 8-Channel 128GB/s High-Bandwidth Memory (HBM) Stacked DRAM with Effective Microbump I/O Test Methods Using 29nm Process and TSV," *ISSCC*, pp. 432-433, 2014.
[3] K. Sohn et al., "A 1.2V 20nm 307GB/s HBM DRAM with At-Speed Wafer-Level I/O Test Scheme and Adaptive Refresh Considering Temperature Distribution," *ISSCC*, pp. 316-317, 2016.
[4] J. H. Cho et al., "A 1.2V 64Gb 341GB/s HBM2 Stacked DRAM with Spiral Point-to-Point TSV Structure and Improved Bank Group Data Control," *ISSCC*, pp. 208-209, 2018.
[5] C.-S. Oh et al., "A 1.1V 16GB 640GB/s HBM2E DRAM with a Data-Bus Window-Extension Technique and a Synergetic On-Die ECC Scheme," *ISSCC*, pp.330-331, 2020.
[6] D. Lee et al., "A 128Gb 8-High 512GB/s HBM2E DRAM with a Pseudo Quarter Bank Structure, Power Dispersion and an Instruction-Based At-Speed PMBIST," *ISSCC*, pp. 334-335, 2020.
[7] H. Shin et al., "McDRAM: Low Latency and Energy-Efficient Matrix Computations in DRAM," *IEEE TCADICS*, vol. 37, no. 11, pp. 2613-2622, Nov. 2018.
[8] F. Devaux, "The True Processing in Memory Accelerator," *IEEE Hot Chips Symp.*, pp. 1-24, 2019.
[9] S. Kim et al., "Charge-Aware DRAM Refresh Reduction with Value Transformation," *IEEE HPCA*, pp. 663-676, 2020.

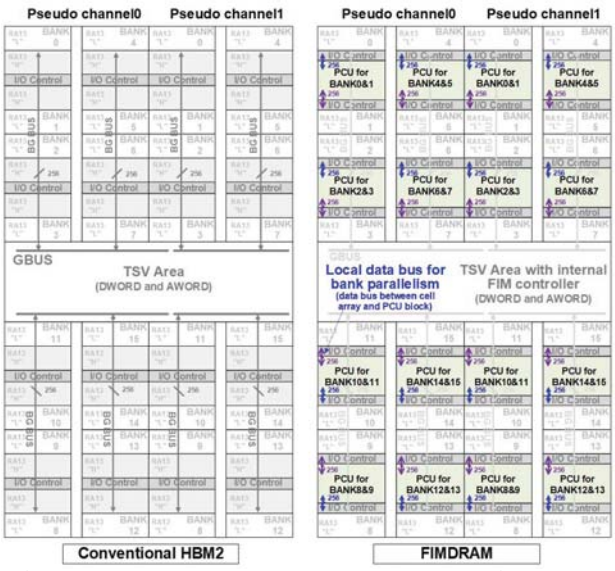

Figure 25.4.1: Architecture of FIMDRAM based on HBM2 compared to a conventional HBM2 for 1CH.

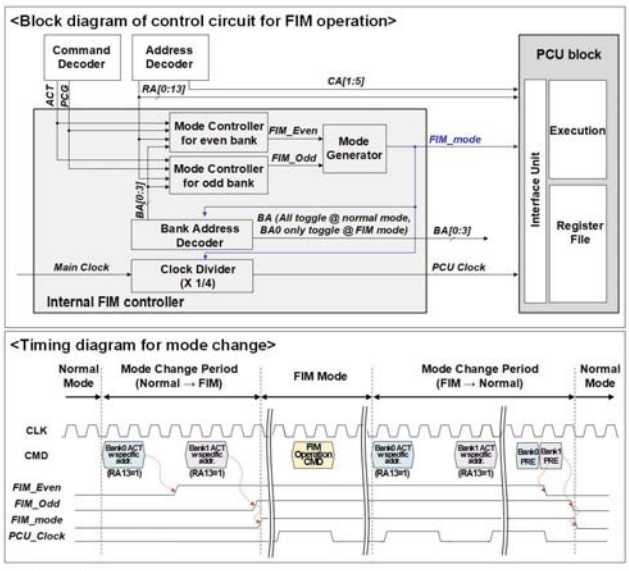

Figure 25.4.2: Mode change method and timing diagram for operation of FIMDRAM.

<Data movement according to operation mode>

Index	Mode	CMD	RA13	Data Movement
①	Normal	WR	L	Data write to cell array
②		WR	H	Not available
③		RD	L	Data read to cell array
④		RD	H	Not available
⑤	FIM	WR	L	Data movement from PCU block to cell array
⑥		WR	H	Data write to PCU register
⑦		RD	L	Data movement from cell array to PCU block
⑧		RD	H	Not available

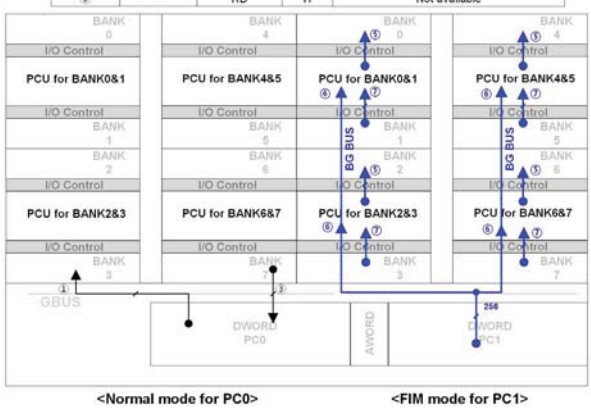

Figure 25.4.3: Data movement and multi bank operation in accordance with FIM mode.

<Available instruction list for FIM operation>

Type	CMD	Description
Floating Point	ADD	FP16 addition
	MUL	FP16 multiplication
	MAC	FP16 multiply-accumulate
	MAD	FP16 multiply and add
Data Path	MOVE	Load or store data
	FILL	Copy data from bank to GRFs
Control Path	NOP	Do nothing
	JUMP	Jump instruction
	EXIT	Exit instruction

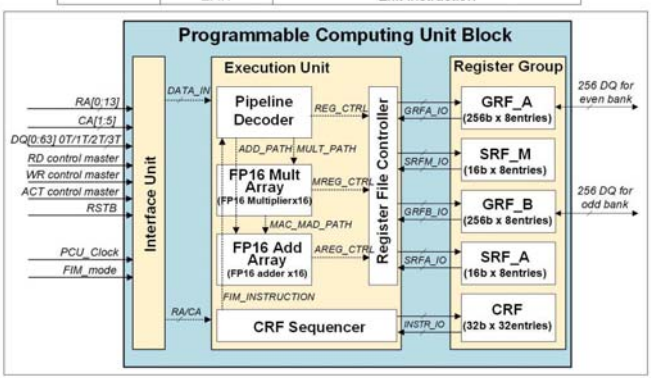

Figure 25.4.4: Block diagram and available instruction list of PCU block.

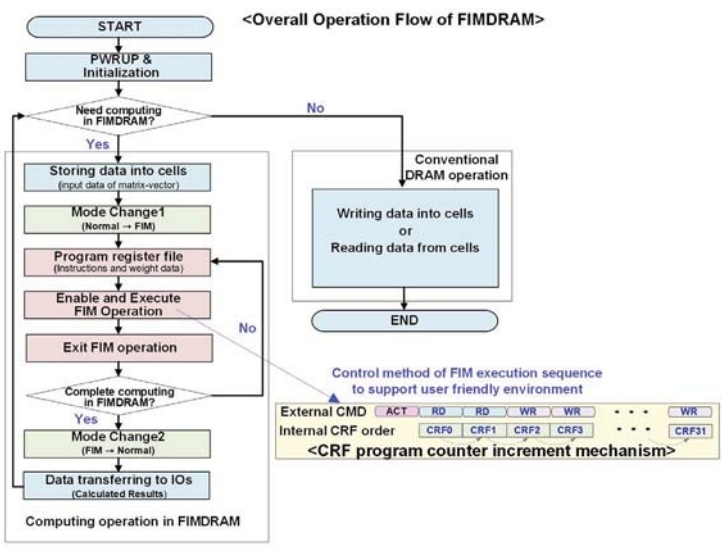

Figure 25.4.5: Operation flow for data computing of FIMDRAM.

<Key Feature Summary>

	[3]	[7]	[8]	FIMDRAM (this work)
Type of DRAM	HBM2	LPDDR4	DDR4	HBM2
Process	20 nm	20 nm	2x nm	20 nm
Memory density	8GB/cube (Buffer-die + 8H 8Gb core-die)	8GB/chip (8H 8Gb mono die)	8GB/DIMM	6GB/cube (Buffer-die + 4H 4Gb core-die with PCU + 4H 8Gb core-die)
Data rate	2.4Gbps	3.2Gbps	2.4Gbps	2.4Gbps
Bandwidth	307GB/s per cube	25.6GB/s per chip	19.2GB/s per DIMM	307GB/s per cube
# of CH	8 per cube	1 per chip	16 per DIMM	8 per cube
# of processing unit	No	2048 per chip (256 per die)	128 per DIMM (8 per chip)	128 per cube (32 per core-die)
Processing operation speed	-	250Mhz @simulation	500MHz @ Measurement	300MHz @ Measurement
Peak throughput	-	0.5 TOPS per chip (250MHz x 256 x 8byte)	0.5 TOPS per DIMM (500MHz x 128 x 8byte)	1.2 TFLOPS per cube (300MHz x 128 x 32byte)
Operation Precision	-	INT8	INT8	FP16

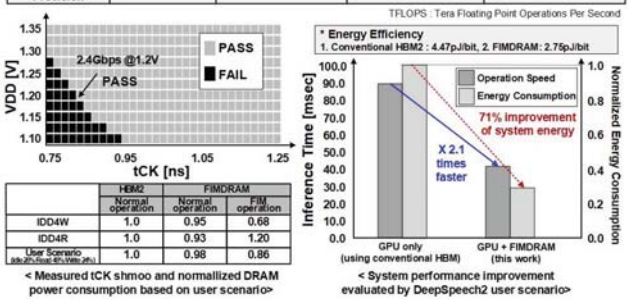

Figure 25.4.6: Measurement results and system performance improvements based on application benchmarks.

25

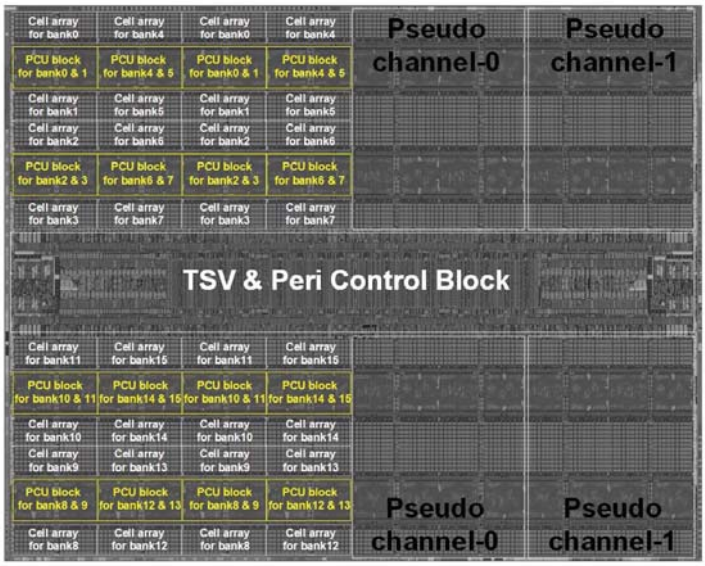

Figure 25.4.7: Chip micrograph of FIMDRAM with PCU block fabricated in the 20nm DRAM process.

978-1-7281-9549-0/21/$31.00 ©2021 IEEE