Verifying DRAM Addressing in Software

Martin Heckel[1,2], Florian Adamsky[2], Jonas Juffinger[1],
Fabian Rauscher[1], and Daniel Gruss[1]

September 23, 2025

[1] Graz University of Technology
[2] Hof University of Applied Sciences

NeRAM
FWF Der Wissenschaftsfonds.
DFG Deutsche Forschungsgemeinschaft

Outline

Background

DRAM Addressing Function Reverse-Engineering

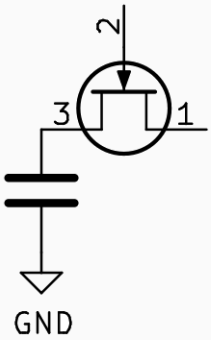
DRAM Addressing Function Verification

Row-Conflict Covert Channel on DDR5

Background

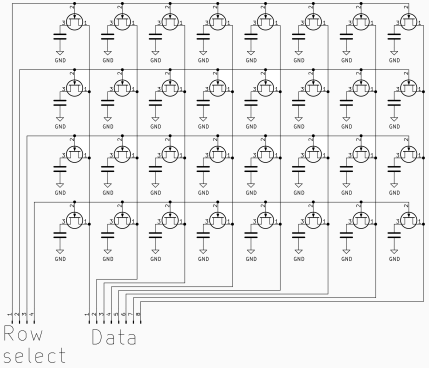
DRAM – Cells

- A single cell consists of:
  - Capacitor storing the data in form of electric charge
  - Transistor controlling the access to the capacitor
- Reading procedure: Enable the control pin and read the voltage at the access pin
- Writing procedure: Apply the level that should be written to the access pin and enable the control pin

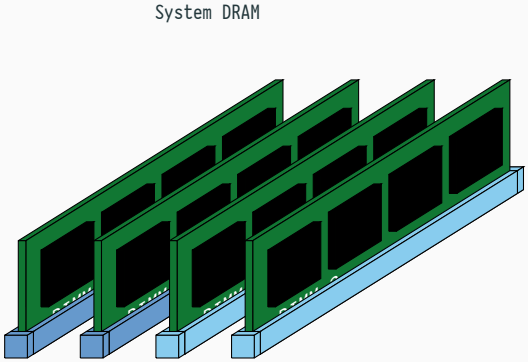


DRAM – Array

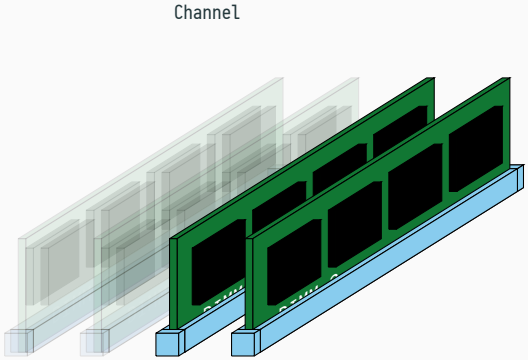
- Multiple cells are organized in an array
- Control pins of the cells connected in rows (only entire rows can be enabled)
- Access pins of the cells conneted in columns
- Capacitors loose chage over time, so it is required to refresh the cells periodically (by default 64 ms for DDR3 and DDR4, 32 ms for DDR5)



DRAM – physical architecture

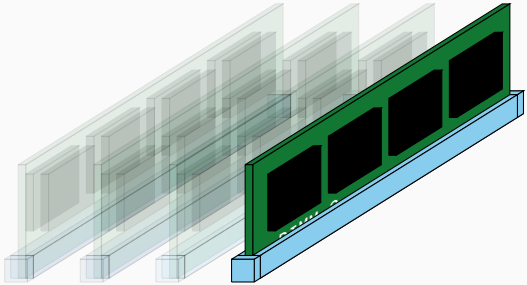


DRAM – physical architecture

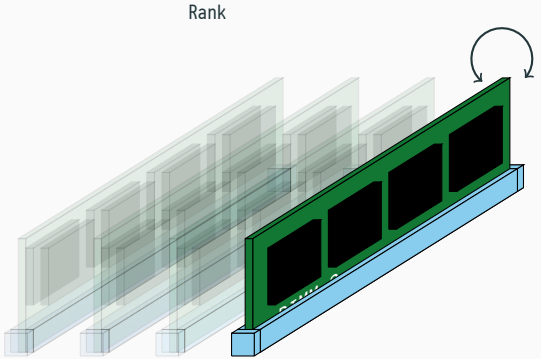


DRAM – physical architecture

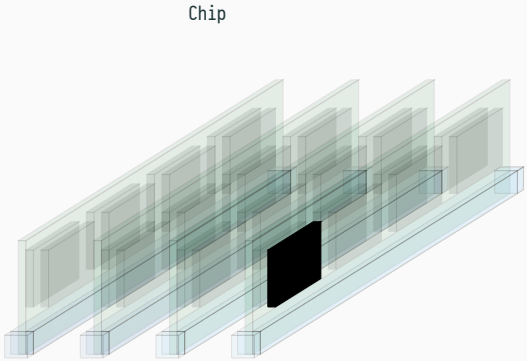
DIMM



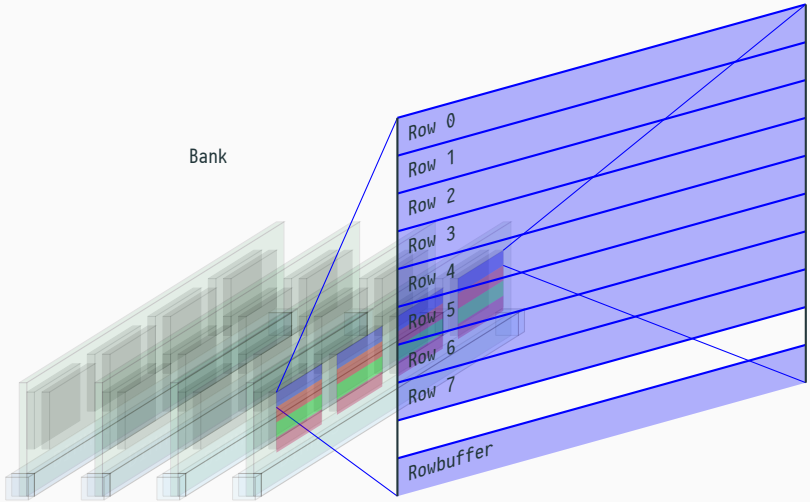
DRAM – physical architecture



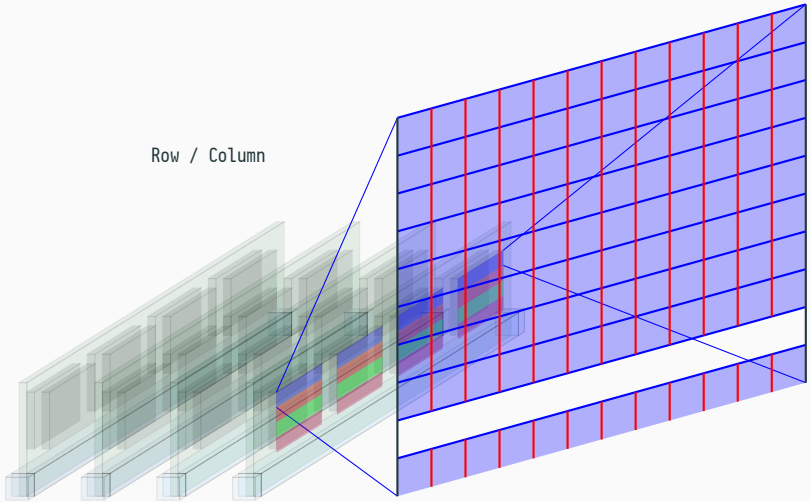
DRAM – physical architecture



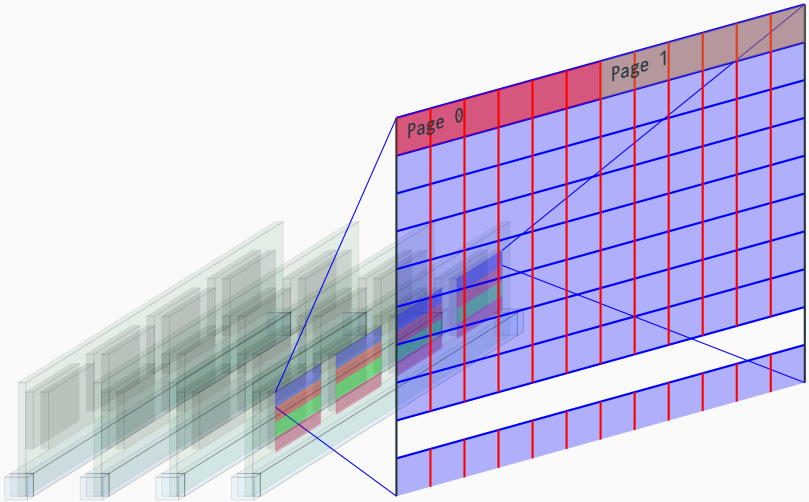
DRAM – physical architecture



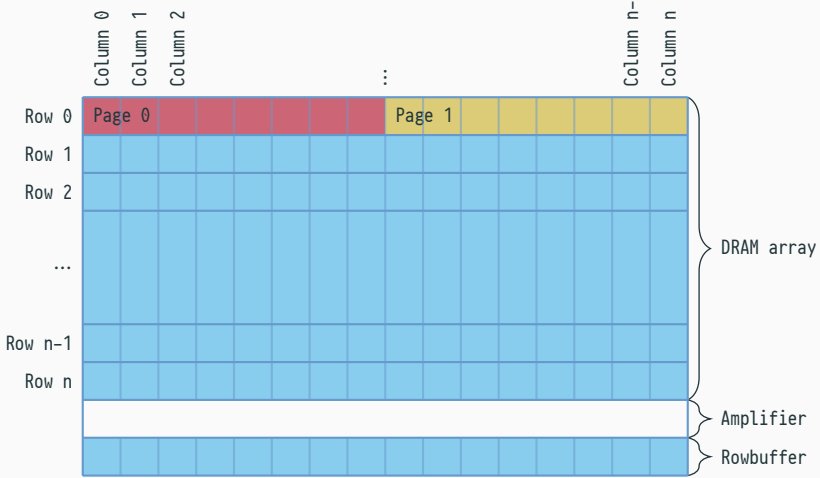
DRAM – physical architecture



DRAM – physical architecture



Structure within a DRAM bank



DRAM Addressing

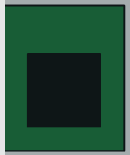
- Data is stored in physical memory:
  - Channel
  - DIMM
  - Rank
  - Bank
  - Row
  - Column
- The Memory Controller translates physical addresses to memory locations

DRAM Addressing

- Data is stored in physical memory:
  - C
  - D
  - R
  - B
  - R
  - C
- The *M*
  addresses to memory locations

decode-dimms

```
---=== Memory Characteristics ===---
Maximum module speed                    1333 MT/s (PC3-10600)
Size                                    4096 MB
Banks x Rows x Columns x Bits           8 x 15 x 10 x 64
Ranks                                   2
```



DRAM Addressing Functions

Physical Address (32 bit, $2^{32} = 4$ GiB)

DRAM Addressing Functions

Physical Address (32 bit, $2^{32} = 4\,\text{GiB}$)

$2^{15} = 32\,768$ Rows

DRAM Addressing Functions

Physical Address (32 bit, $2^{32} = 4\,\text{GiB}$)

$2^{15} = 32\,768$ Rows

$2^{10} = 1024$ Columns

DRAM Addressing Functions

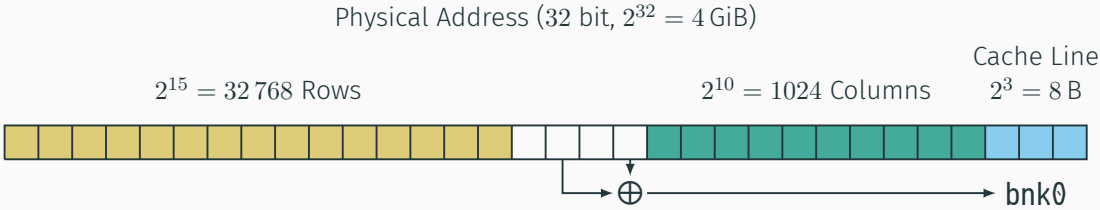
Physical Address (32 bit, $2^{32} = 4$ GiB)

$2^{15} = 32\,768$ Rows

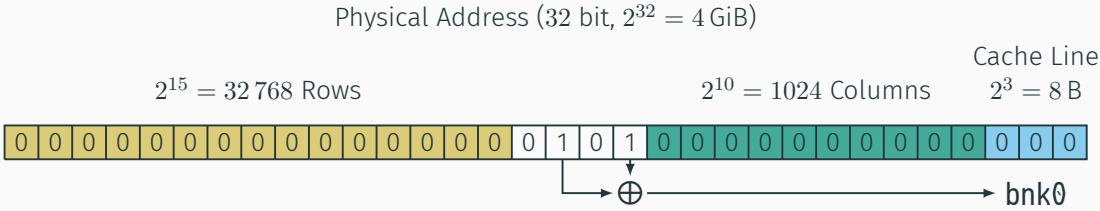
$2^{10} = 1024$ Columns

Cache Line
$2^3 = 8$ B

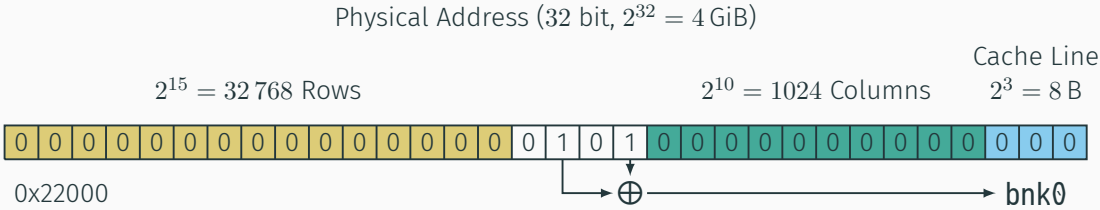
DRAM Addressing Functions



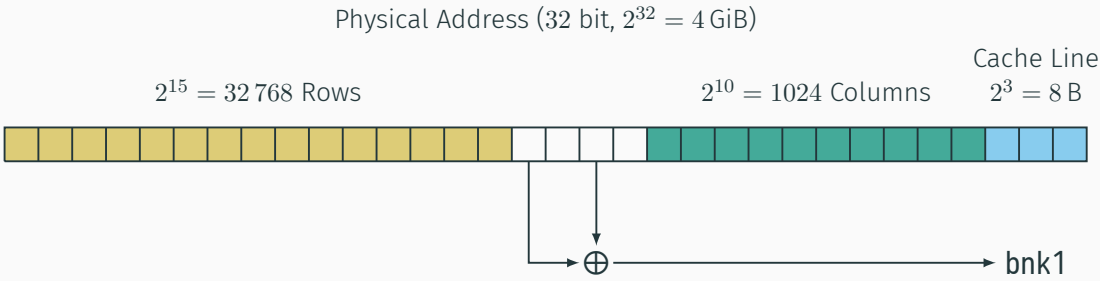
DRAM Addressing Functions



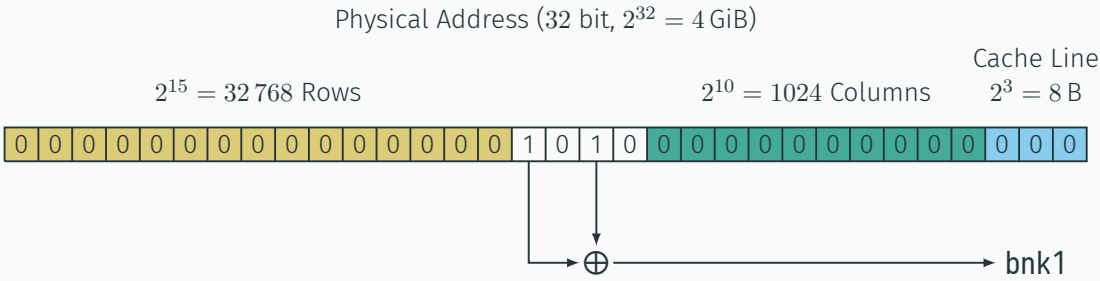
DRAM Addressing Functions



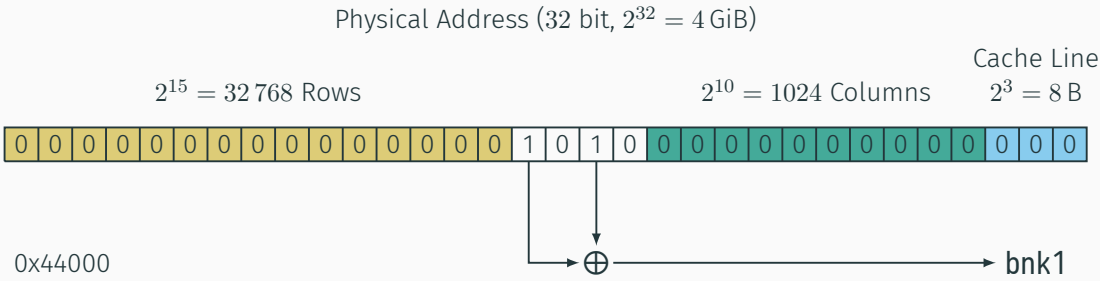
DRAM Addressing Functions



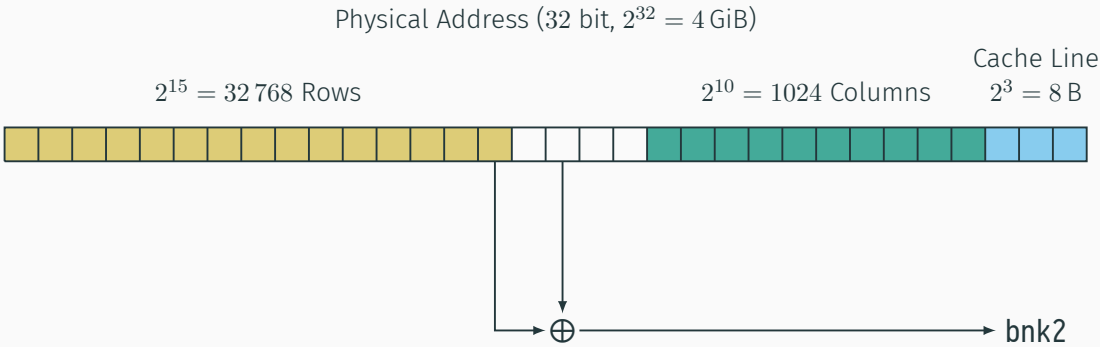
DRAM Addressing Functions



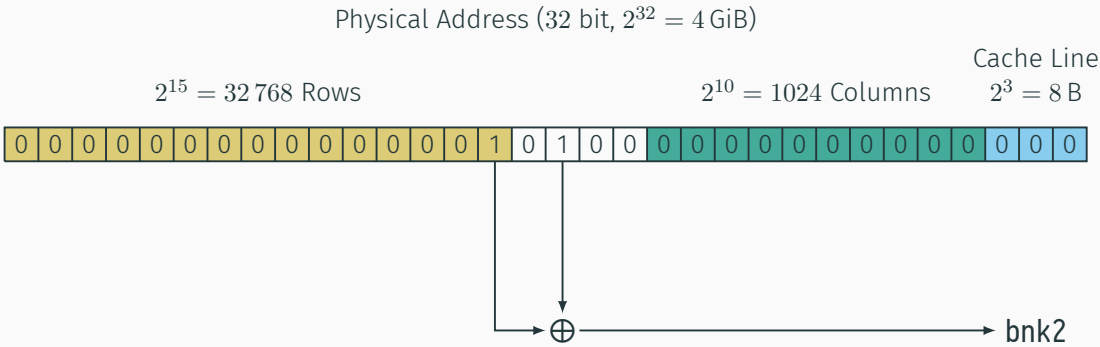
DRAM Addressing Functions



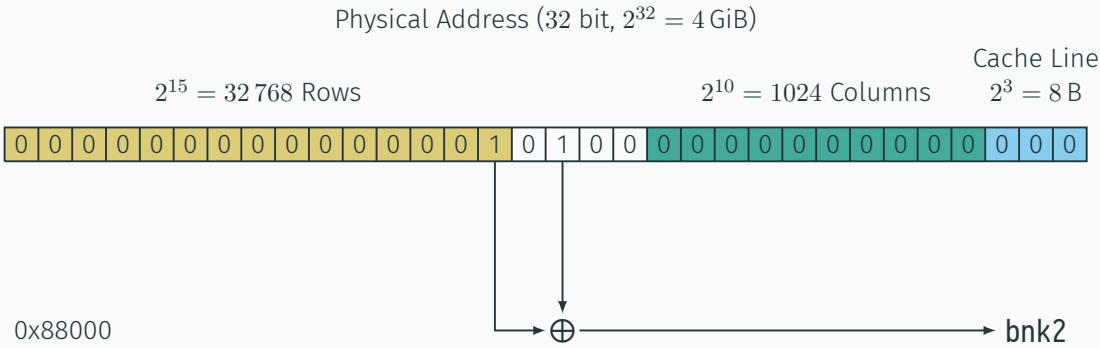
DRAM Addressing Functions



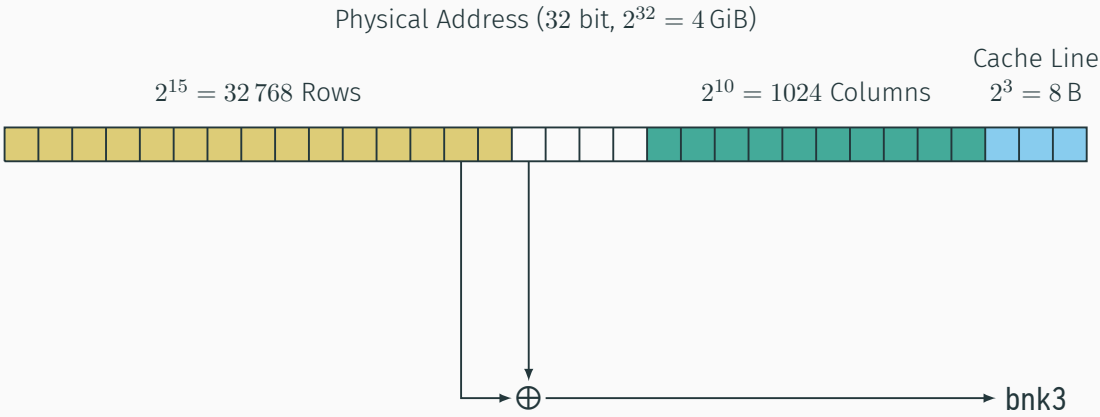
DRAM Addressing Functions



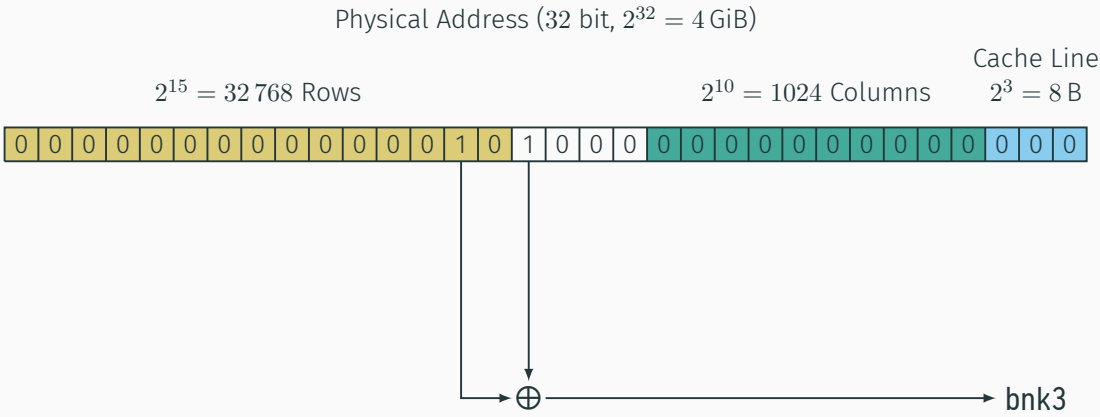
DRAM Addressing Functions



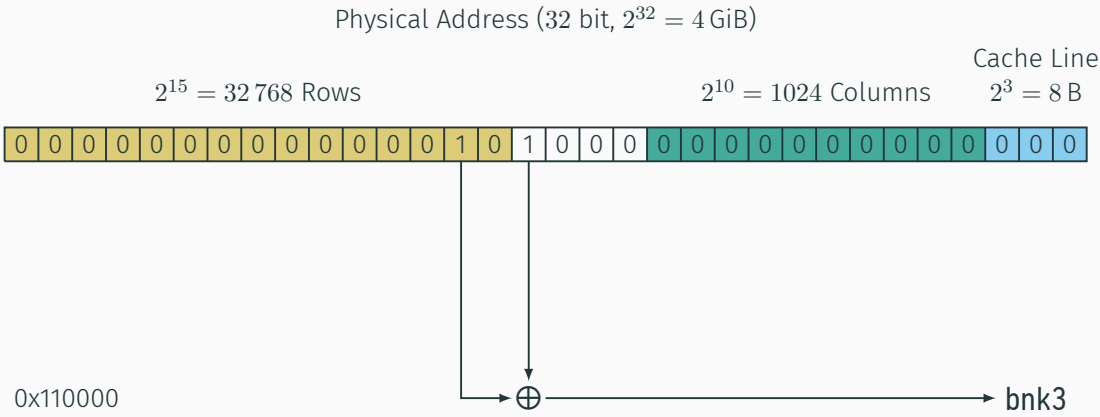
DRAM Addressing Functions



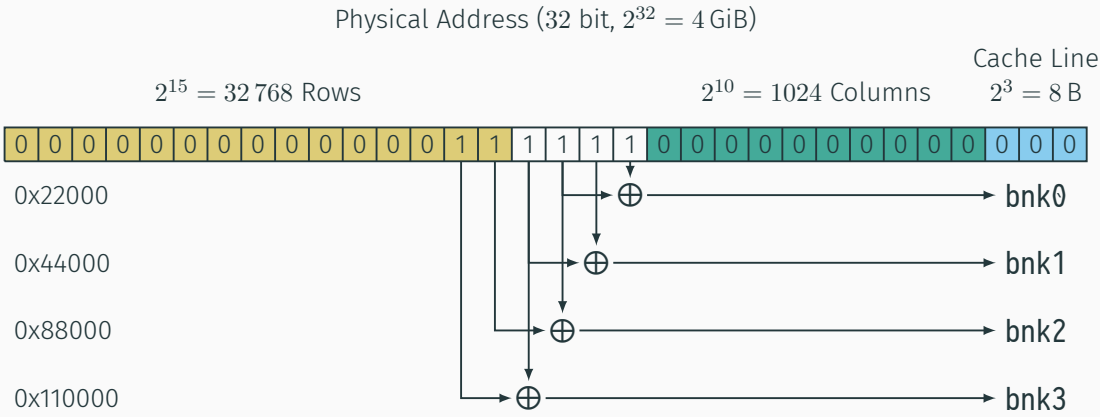
DRAM Addressing Functions



DRAM Addressing Functions



DRAM Addressing Functions

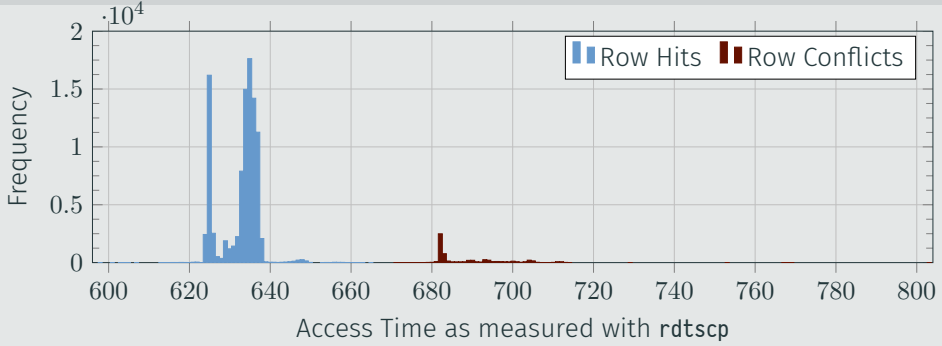


DRAM Addressing Function Reverse-Engineering

DRAM Access Times

- The row buffer is shared among all rows in one bank
- Reading a row from the DRAM array is destructive
- Accesses to different rows at the same bank require the content of the row buffer to be restored to the DRAM array first (*Row Conflict*)
- Accesses to the same row at the same bank does not require restoring the row buffer and loading another row (*Row Hit*)
- Row Hits are faster than Row Conflicts

DRAM Access Times



Grouping Addresses based on DRAM Access Times

- Find threshold between Row Hits and Row Conflicts
- Access addresses alternatingly and compare access time t to the threshold t_T
  - $t < t_T \Rightarrow$ Row Hit
  - $t > t_T \Rightarrow$ Row Conflict
- Idea: Group addresses with Row Conflicts together

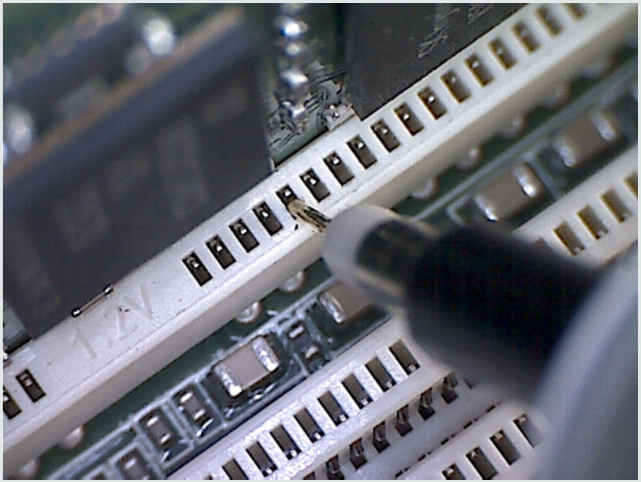
DRAM Addressing Function Reverse-Engineering

- Idea: Identify addressing functions that separate the groups based on all physical addresses in these groups
- Solved for linear DRAM addressing functions by Pessl et al. [1]
- Still not solved for nonlinear DRAM addressing functions to the best of our knowledge

Problem: Verification of DRAM Addressing Functions

- The DRAM addressing functions reverse-engineered by a specific tool might not be correct
- Different tools might return different functions
- Typical verification: Try to use the reverse-engineered functions for a Rowhammer attack
- Problem: Rowhammer works also with random accesses as shown by Seaborn and Dullien [2]
- **How can we verify DRAM addressing function correctness?**

Verification with Physical Probing as used by Pessl et al. [1]

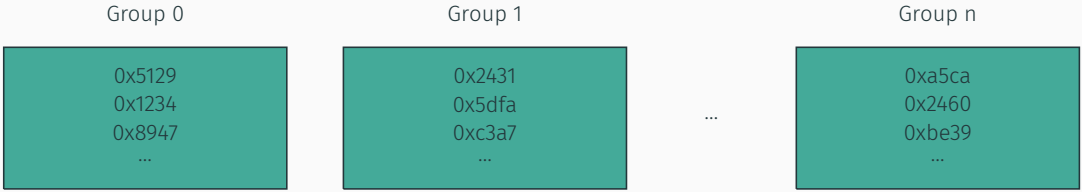


DRAM Addressing Function Verification

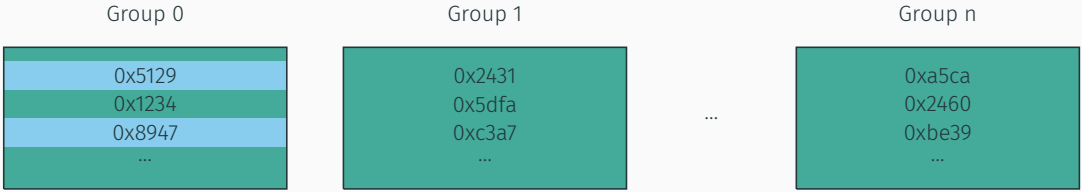
Idea: Assume correct Functions and verify this Assumption

- Group DRAM addresses based on the DRAM addressing functions that should be verified
- If correctly grouped:
  - Row Hits should be measured between randomly selected addresses of different groups
  - Row Conflicts should be measured between randomly selected addresses of the same group

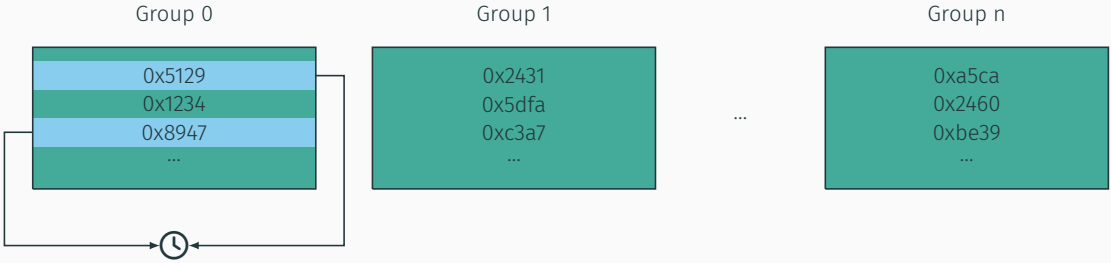
Idea: Assume correct Functions and verify this Assumption



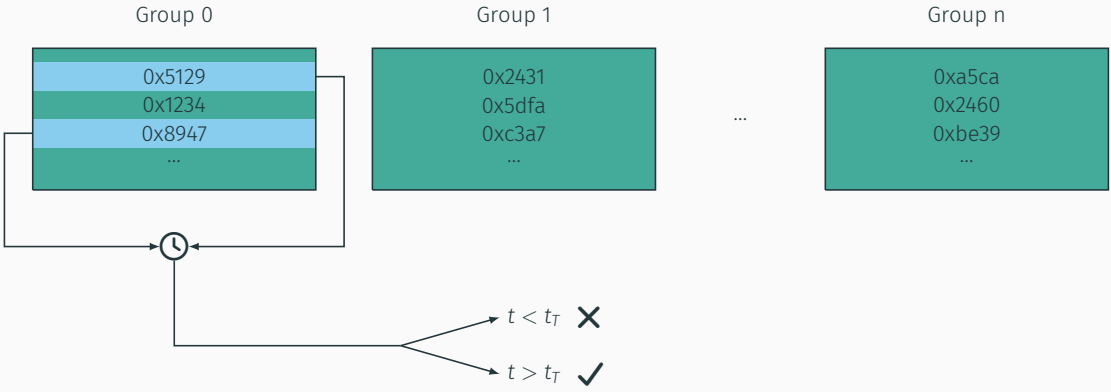
Idea: Assume correct Functions and verify this Assumption



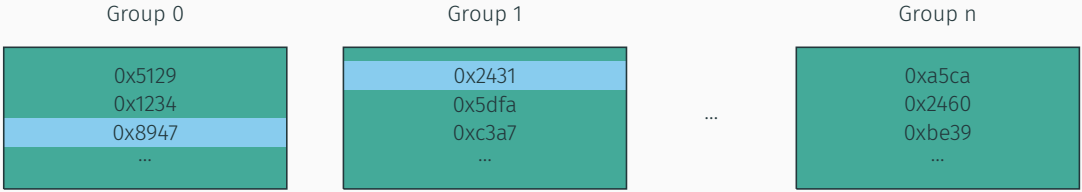
Idea: Assume correct Functions and verify this Assumption



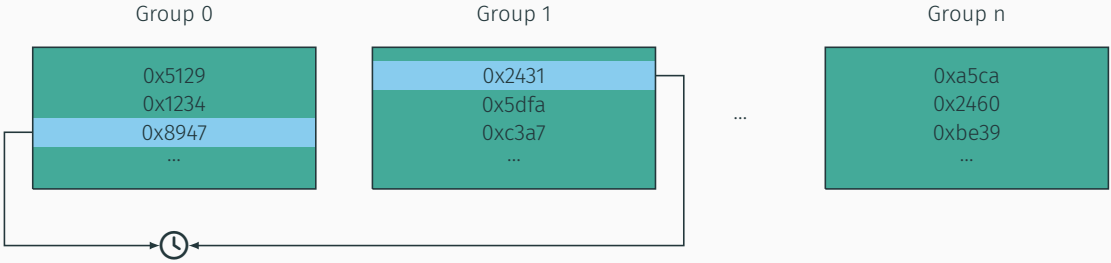
Idea: Assume correct Functions and verify this Assumption



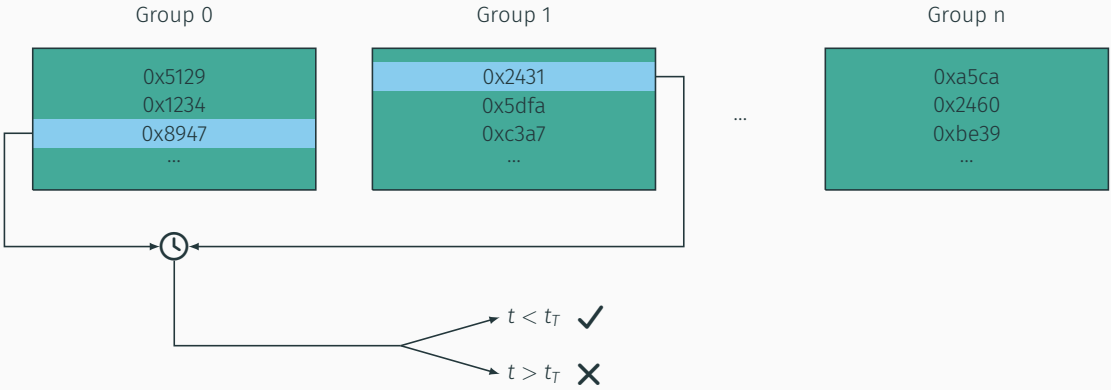
Idea: Assume correct Functions and verify this Assumption



Idea: Assume correct Functions and verify this Assumption



Idea: Assume correct Functions and verify this Assumption

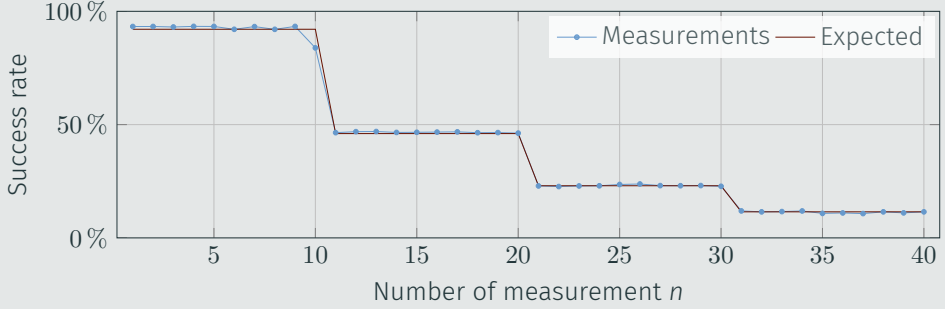


Verify single DRAM Addressing Functions

- Each DRAM bank addressing function returns one bit of the DRAM bank number
- When one function is removed, only half of the banks can be addressed with the remaining functions
- Therefore, the DRAM bank number should be incorrect for half of the addresses (which are still distributed over all banks).

Verify single DRAM Addressing Functions

Overall Correctness when removing Functions



DRAM Addressing Function Reverse-Engineering Tools on DDR3

	PoC	AFn Mask	$\%_{avg}$	σ	$\%_{min}$	$\%_{max}$
S301	AMDRE		83.5 %	27.4	1.4 %	92.9 %
	DRAMDig		92.7 %	0.1	92.5 %	92.8 %
	Drama		43.9 %	34.1	8.3 %	92.7 %
	Dare		0.0 %	0.0	0.0 %	0.0 %
	TRRespass		0.0 %	0.0	0.0 %	0.0 %

	PoC	AFn Mask	$\%_{avg}$	σ	$\%_{min}$	$\%_{max}$
S302	AMDRE		90.1 %	0.1	90.0 %	90.3 %
	DRAMDig		89.8 %	0.8	87.3 %	90.2 %
	Drama		42.3 %	35.4	0.0 %	90.1 %
	Dare		0.0 %	0.0	0.0 %	0.0 %
	TRRespass		0.0 %	0.0	0.0 %	0.0 %

DRAM Addressing Function Reverse-Engineering Tools on DDR4

	PoC	AFn Mask	$\%_{avg}$	σ	$\%_{min}$	$\%_{max}$
401	AMDRE		85.4 %	0.1	85.2 %	85.5 %
	DRAMDIG		84.8 %	0.2	84.4 %	85.1 %
	DRAMA		15.3 %	15.6	0.0 %	44.9 %
	DARE		76.8 %	25.6	0.0 %	85.5 %
	TRRESPASS		84.1 %	2.1	79.6 %	85.6 %
402	AMDRE		77.2 %	0.2	76.9 %	77.4 %
	DRAMDIG		42.3 %	0.1	42.0 %	42.5 %
	DRAMA		6.7 %	6.5	0.0 %	21.3 %
	DARE		29.6 %	19.4	0.0 %	42.4 %
	TRRESPASS		42.2 %	0.1	42.0 %	42.4 %

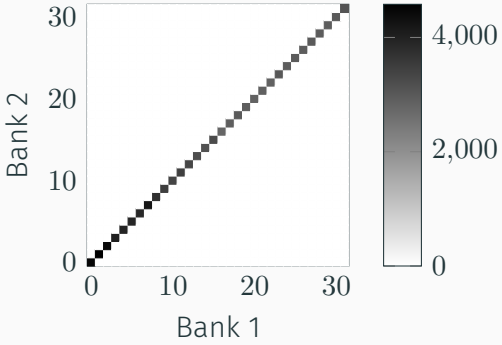
	PoC	AFn Mask	$\%_{avg}$	σ	$\%_{min}$	$\%_{max}$
S403	AMDRE		23.2 %	13.0	0.0 %	38.8 %
	DRAMDIG		0.0 %	0.0	0.0 %	0.0 %
	DRAMA		13.9 %	9.7	0.0 %	23.1 %
	DARE		14.4 %	6.0	0.0 %	21.6 %
	TRRESPASS		0.0 %	0.0	0.0 %	0.0 %
S404	AMDRE		62.3 %	0.5	61.8 %	63.4 %
	DRAMDIG		0.0 %	0.0	0.0 %	0.0 %
	DRAMA		16.0 %	5.8	7.3 %	21.8 %
	DARE		18.5 %	1.6	16.1 %	20.5 %
	TRRESPASS		0.0 %	0.0	0.0 %	0.0 %

DRAM Addressing Function Reverse-Engineering Tools on DDR5

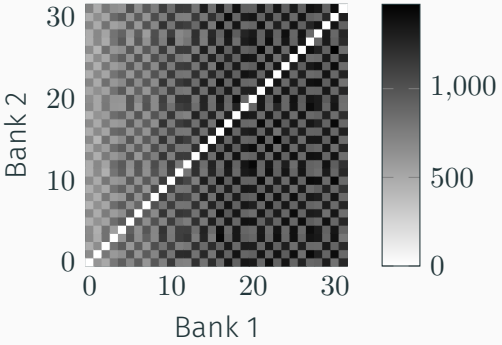
	PoC	AFn Mask	$\%_{avg}$	σ	$\%_{min}$	$\%_{max}$
S501	AMDRE		42.9 %	42.9	0.0 %	86.0 %
	DRAMDIG		0.0 %	0.0	0.0 %	0.0 %
	DRAMA		5.1 %	1.8	1.6 %	6.3 %
	DARE		6.3 %	0.0	6.3 %	6.4 %
	TRRESPASS		5.4 %	1.2	2.9 %	6.1 %
S502	AMDRE		87.0 %	0.2	86.7 %	87.3 %
	DRAMDIG		0.0 %	0.0	0.0 %	0.0 %
	DRAMA		4.0 %	2.2	0.0 %	5.5 %
	DARE		4.6 %	1.7	0.0 %	5.4 %
	TRRESPASS		5.4 %	0.1	5.2 %	5.5 %

	PoC	AFn Mask	$\%_{avg}$	σ	$\%_{min}$	$\%_{max}$
S503	AMDRE		0.0 %	0.0	0.0 %	0.0 %
	DRAMDIG		0.0 %	0.0	0.0 %	0.0 %
	DRAMA		22.7 %	0.6	21.8 %	23.7 %
	DARE		1.2 %	1.2	0.0 %	2.6 %
	TRRESPASS		0.0 %	0.0	0.0 %	0.0 %
S504	AMDRE		0.0 %	0.0	0.0 %	0.0 %
	DRAMDIG		0.0 %	0.0	0.0 %	0.0 %
	DRAMA		20.6 %	6.9	0.0 %	23.6 %
	DARE		18.9 %	9.5	0.0 %	23.7 %
	TRRESPASS		0.0 %	0.0	0.0 %	0.0 %

Number of wrongly classified DRAM banks



(a) Too fast Row Conflicts



(b) Too slow Row Hits

Rank Addressing Functions

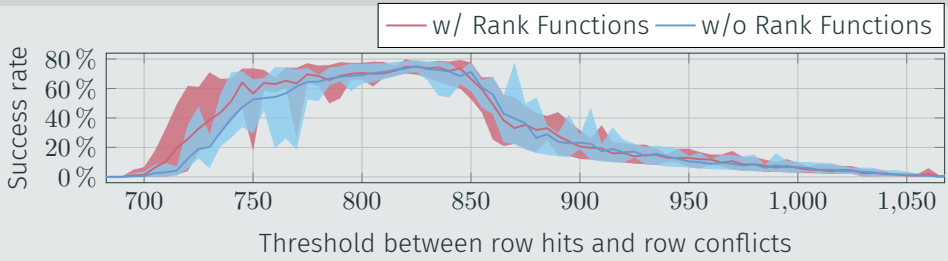
- Some banks have more too slow Row Hits than others
- The pattern can be described with the linear function `0x0d` and is applied to the numbers of banks
- The system has two ranks and one DIMM
- The effect only occurs on systems when the DIMM has two ranks
- We assume that this is related to the rank select commands

Experimental Evaluation of Rank Addressing Functions

- We selected the other group from which we take the address to measure the row hit in a way it is on the same rank
- Thereby, we see an increase in the success rate when applying rank functions
- This effect only happens when the threshold is selected in a specific range
- Otherwise, the row hits are either always classified correctly (threshold higher) or always classified wrong (threshold lower)

Experimental Evaluation of Rank Addressing Functions

DRAM Rank Addressing Functions



Row-Conflict Covert Channel on DDR5

Row-Conflict Covert Channel

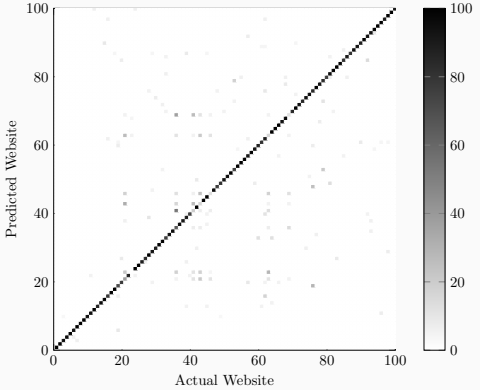
- Pessl et al. [1] introduced a covert channel based on the timing difference between Row Hits and Row Conflicts
- We re-implemented their approach and verified it on DDR3 (up to $2.23\,\text{Mbit}\,\text{s}^{-1}$), DDR4 (up to $0.66\,\text{Mbit}\,\text{s}^{-1}$), and DDR5 (up to $1.39\,\text{Mbit}\,\text{s}^{-1}$)
- We implemented cross-VM synchronization, which enables the covert channel to work cross-VM on the same host

Website Fingerprinting Attack

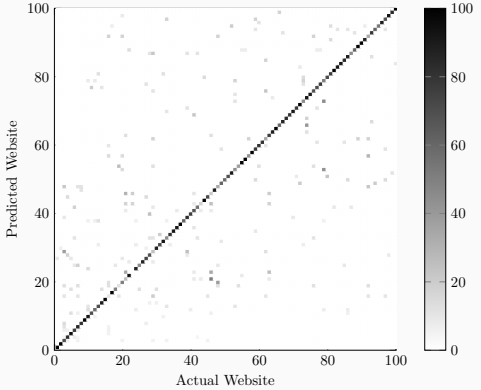
- We utilized the covert channel described before to perform a website fingerprinting attack
- We measure the access times to DRAM addresses of different banks (one thread per bank, $n_{\text{proc}} - 2$ threads on a system with n_{proc} logical CPUs)
- Next, a specified windows size ($100\,\mu\text{s}$) is used and the number of row conflicts in that window is stored for each measured bank
- We performed 100 accesses to every website and used a ML model to predict 100 websites ($80\,\%$ of the data for training, $20\,\%$ for validation).

Website Fingerprinting Attack – Experimental Evaluation

- We reached an F_1 score of $84\,\%$ on DDR4 and an F_1 score of $74\,\%$ on DDR5



(a) Website confusion matrix on DDR4, with F_1 score $84.0\,\%$.



(b) Website confusion matrix on DDR5, with F_1 score $74.0\,\%$.

Conclusion

- We have shown a time-based approach to verify single DRAM bank addressing functions without physical probing
- We have reverse-engineered Rank Addressing Functions
- We reproduced the Covert Channel from Pessl et al. [1] on DDR5
- We utilized that Covert Channel to perform a Website Fingerprinting Attack on DDR4 and DDR5

Verifying DRAM Addressing in Software

Martin Heckel[1,2], Florian Adamsky[2], Jonas Juffinger[1],
Fabian Rauscher[1], and Daniel Gruss[1]

September 23, 2025

[1] Graz University of Technology
[2] Hof University of Applied Sciences

NeRAM
FWF Der Wissenschaftsfonds.
DFG Deutsche Forschungsgemeinschaft