HYU

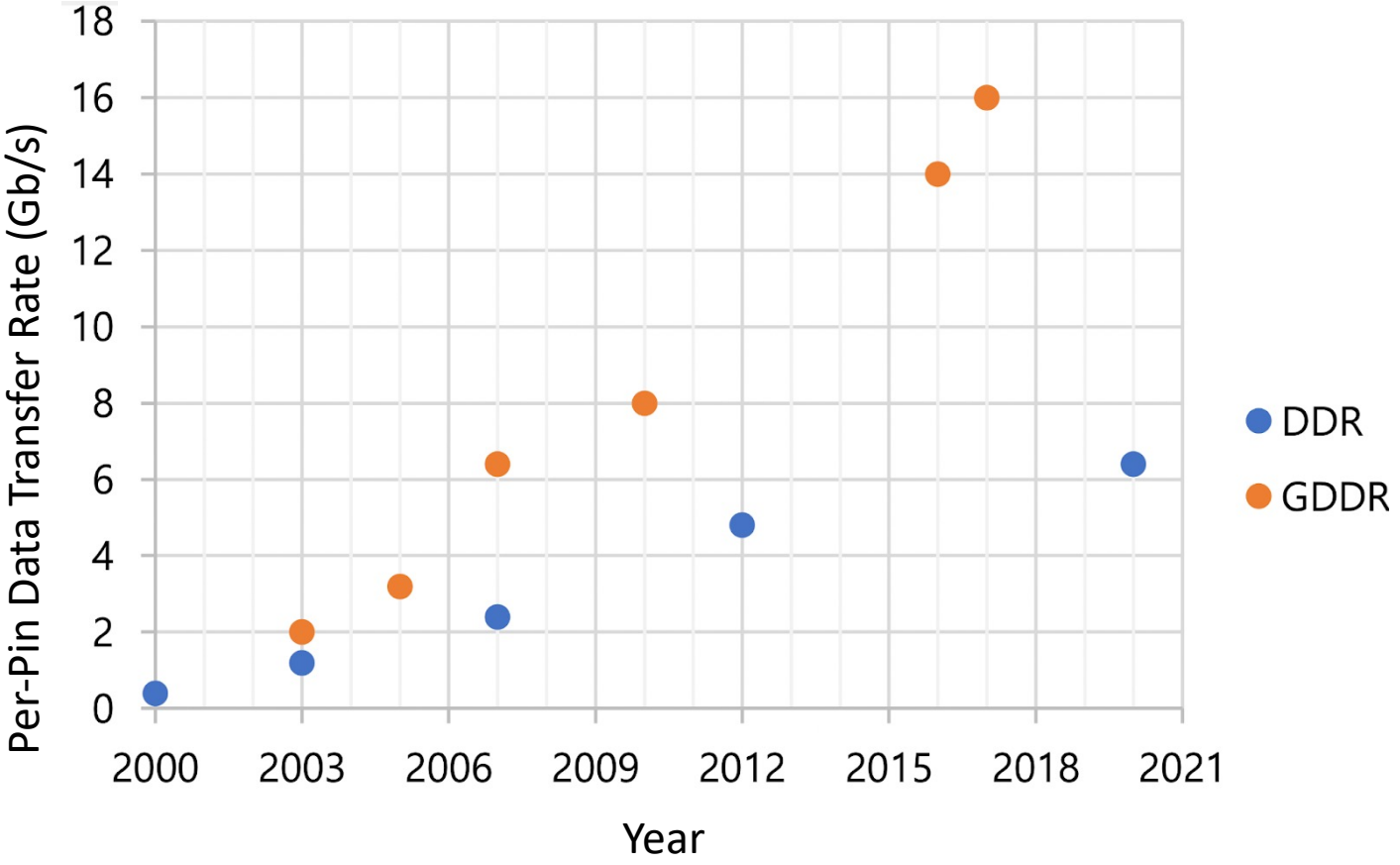
Energy-Efficient Bus Encoding Techniques for Next-Generation PAM-4 DRAM Interfaces

Youri Su, Sanghun Lee, Eunji Song,
Dongha Kim, Jaeduk Han, and Hokeun Kim

Dept. of Electronic Engineering, Hanyang University
IoT Lab & Nifty Chips Lab

DRAM BW Trends

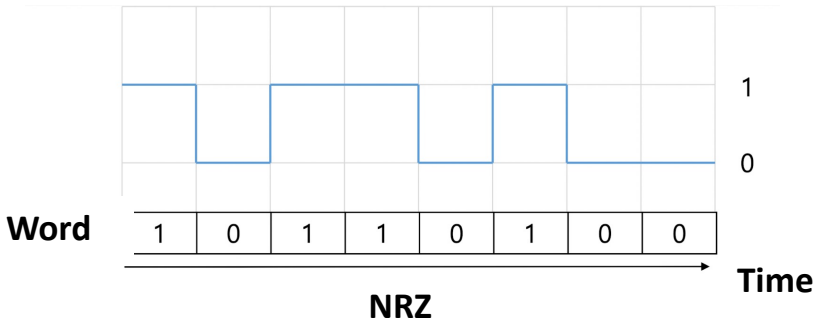
- Per-pin data rates of computing and graphic DRAM interface increase 2x every 3-4 years

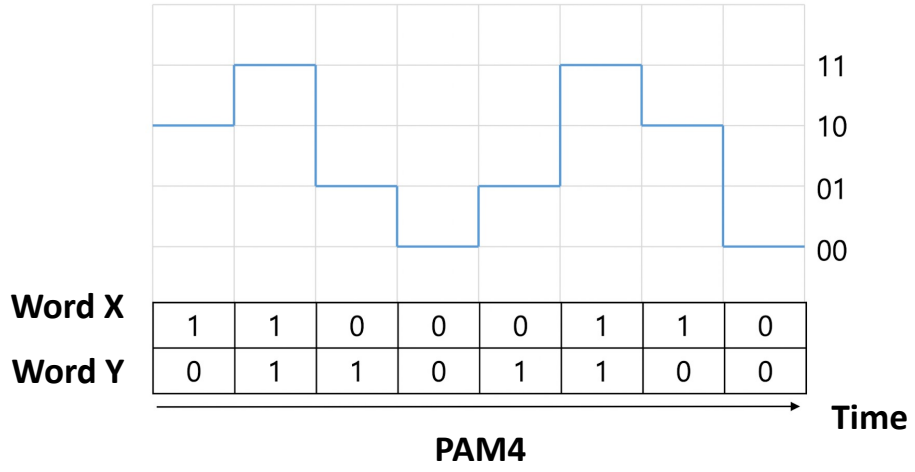


Signal Modulation

- Channel BW is limited because of the physical limitation of PCB
- PAM-4* sends 2 bits per symbol
  - Reduce the Nyquist Frequency for the same data-rate
  - Widely adopted in recent high-speed interfaces

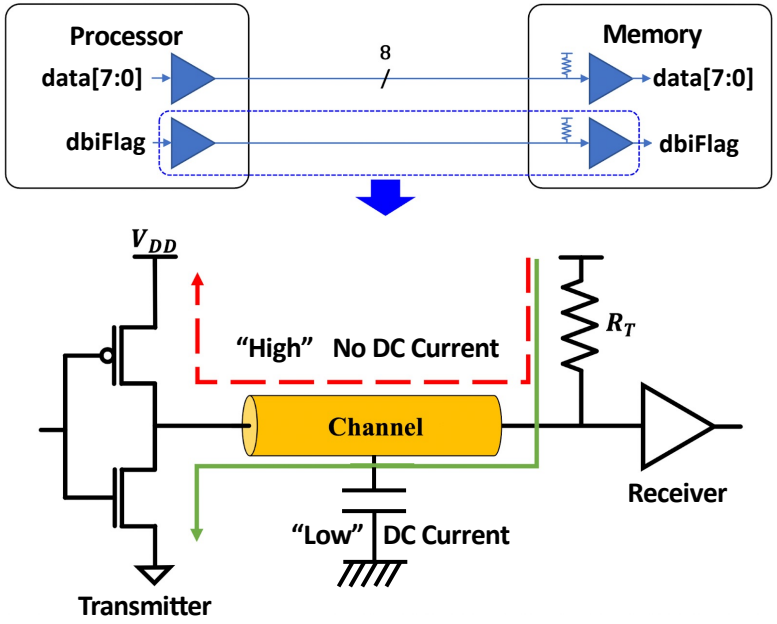
*PAM-4 stands for Pulse Amplitude Modulation 4-level





Power Consumption and Channel Structure

- **Power consumption**
  1) **Termination power**: static current flow through termination R
  2) **Switching power**: energy for charging the channel capacitance
- **POD(Pseudo Open Drain) Termination**
  - No DC current (zero power consumption) when data is high
- **DBI:** (Conditionally) inverts symbols to reduce the termination power
  - Based on the # of zeros with the DBI flag

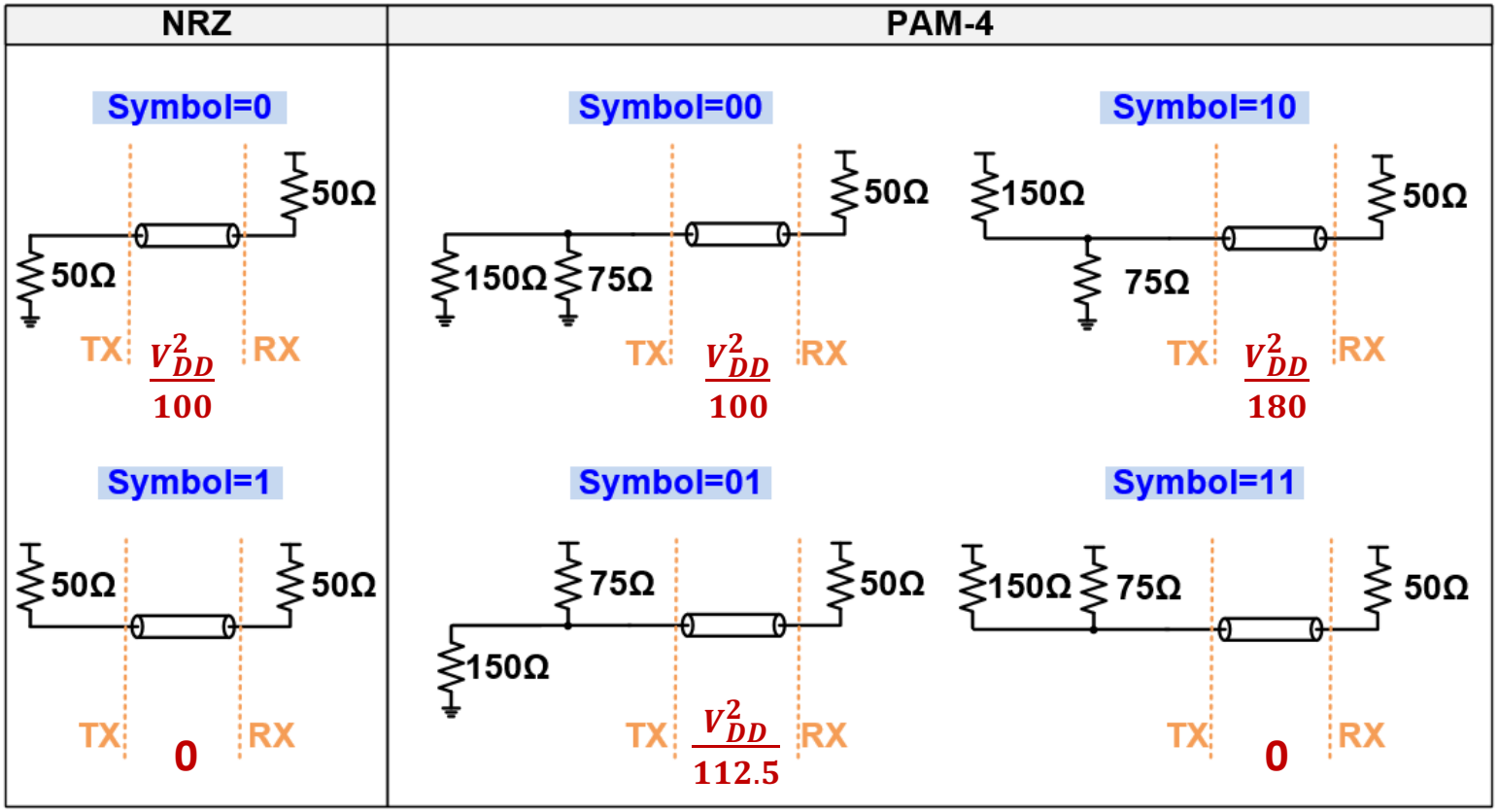


[DBI Table in NRZ]

Data	Data Encoded	DBI Flag	# of Zeros
00000000	11111111	1	0
00000001	11111110	1	1
00000011	11111100	1	2
00000111	11111000	1	3
00001111	11110000	1	4
00011111	11100000	0	4
00111111	11000000	0	3
01111111	10000000	0	2
11111111	00000000	0	1

Motivation to Modify DBI for PAM-4 Signaling

- PAM-4 signaling has 4 different levels of power consumption
- Conventional DBI is inadequate for the PAM-4 signaling



Related Work

CDR[9]	• Uses Software & Hardware to adjust the order of data • Reduces termination power • NRZ
AWR[10]	• Dynamic reordering of data words (NN algorithm) • Reduces switching power • NRZ
STFL-DDR[11]	• Uses a high-performance clock rate in low-power wires • Reduces power consumption
SMOREs[12]	• Utilize the idle period using longer energy-efficient code • Reducing switching power • PAM4

- **Our proposed Encoding methods**
  - Reducing **termination power** on **PAM-4 signaling**
  - **Moderate overhead**, without affecting the transmission rate and latency

[9] B. Feinberg et al., 2020 "Commutative data reordering: a new technique to reduce data movement energy on sparse inference workloads"
[10] E. Maragkoudaki et al., 2020 "Energy-efficient time-based adaptive encoding for off-chip communication"
[11] P. Behnam et al., 2020 "STFL-DDR: Improving the energy- efficiency of memory interface"
[12] M. O'Connor et al., 2022 "Saving PAM4 bus energy with SMOREs: Sparse multi-level opportunistic restricted encodings"

Reducing Transfer Energy with Bus Data Encoding

Encoding Algorithms for Reducing **Termination Power**

- NRZ (Not PAM4)
  - Algorithm 1: NRZ-DBI
- PAM4
  - Algorithm 2: PAM4-DBI (Extension of Algorithm 1)
  - Algorithm 3: PAM4-MF (New - **Proposed**)
  - Algorithm 4: PAM4-Sort (New - **Proposed**)

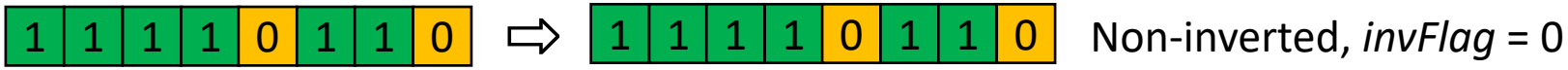
Algorithm 1 NRZ-DBI

- Symbol **1** consumes **less** power
- **Invert** if the **# of 1s ≤ 4** ⇨ Return with **invFlag**

```
1:  NRZ_DBI_ENCODING(oneWord)   ⇨ oneWord: 8-bit single word
2:      invFlag = 0
3:      if sum(oneWord) ≤ 4  then      ⇨ # of 1s ≤ 4
4:          INVERT(oneWord)
5:          invFlag = 1
6:  return oneWord, invFlag
```

Ex)

(# of 0s < # of 1s)

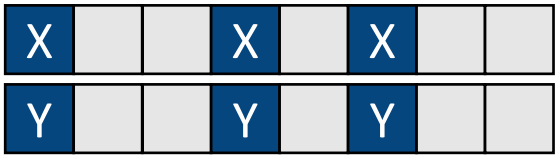


(# of 0s > # of 1s)

Algorithm 2 PAM4-DBI

- **Invert** if $P_{nonDBE} > P_{DBE}$ ⇨ Return with **invFlag**

Ex) Input: 2x8-bit, 2 Words



***cntXY* = 3**

Symbol	00	01	10	11
Power Consumption	$\frac{V_{DD}^2}{100}$	$\frac{V_{DD}^2}{112.5}$	$\frac{V_{DD}^2}{180}$	0

$$\frac{cnt00}{100}+\frac{cnt01}{112.5}+\frac{cnt10}{180}+cnt11\times 0 > \frac{cnt11}{100}+\frac{cnt10}{112.5}+\frac{cnt01}{180}+cnt00\times 0$$

1: **PAM4_DBI_ENCODING**(*twoWords*) ⇨ *twoWords*: 2x8-bit two words
2: *invFlag* = 0
3: **if** $3\times(cnt00-cnt11) > cnt10-cnt01$ **then**
4: INVERT(*twoWords*)
5: *invFlag* = 1
6: **return** *twoWords*, *invFlag*

Algorithm 3 PAM4-MF (Proposed)

- **Count** each symbol's number

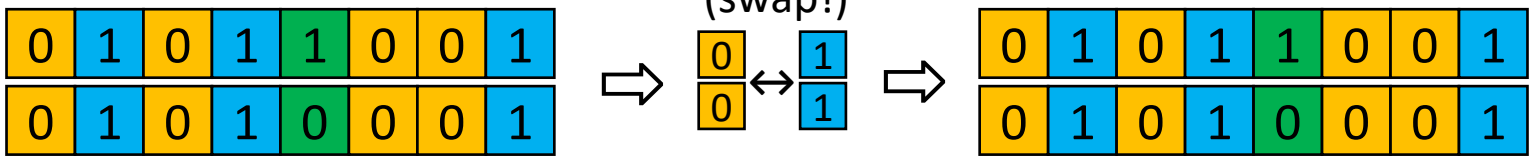
⇨ Find the **Most Frequent** symbol

⇨ **Swap** with **11**

⇨ Return with **Most Frequent symbol** (2-bit)

- Minimal circuit logic overhead

Ex) # of **00**s = 4 (Most Frequent)



1: **PAM4_MF_ENCODING**(*twoWords*) ⇨ *twoWords*: 2x8-bit two words
2: COUNT(*twoWords*) ⇨ Get *cnt00*, *cnt01*, *cnt10*, *cnt11*
3: **MF_CONVERT(twoWords, *Symbol*)**
⇨ *Symbol*: Most Frequent Symbol
⇨ MF_CONVERT: Swap *Symbol* with 11
4: **return** *twoWords*, *Symbol*

Algorithm 4 PAM4-Sort (Proposed)

- **Count** each symbol's number
  ⇨ **Sort** the numbers of each symbol
  ⇨ **Swap** each symbol to match **sorted numbers of symbol**
  ⇨ Return with the **matched permutation number** (4!⇨5-bit)
- **Permutation** number represents the case number

Ex)

0	1	0	1	1	0	0	1
0	1	0	1	0	0	0	1

cnt00 = 4 → 0
cnt01 = 0 → 1
cnt10 = 1 → 3
cnt11 = 3 → 4

⇨

permutation number = 9

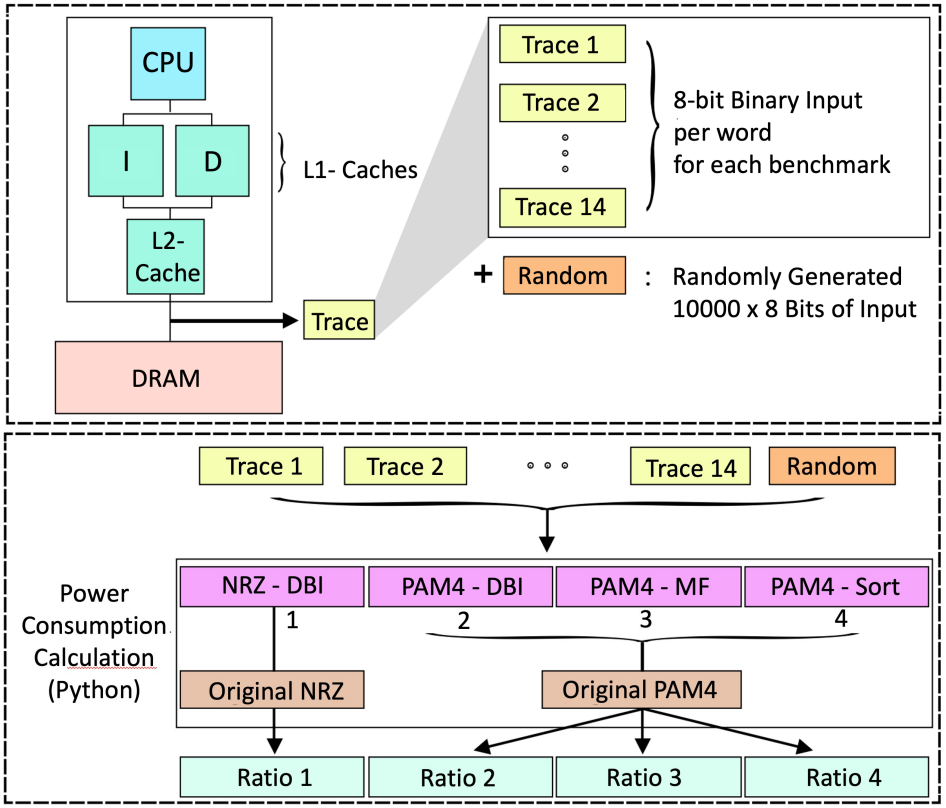
1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	0

1: **PAM4_SORT_ENCODING**(*twoWords*) ⇨ *twoWords*: 2x8-bit two words
2: **SORT(COUNT(*twoWords*))** ⇨ Sort the number of each symbol
3: SORT_CONVERT(*twoWords*, *permutation*)
⇨ SORT_CONVERT: Swap each symbol to match the case
⇨ *permutation*: Represents the case number
4: **return** *twoWords*, *permutation*

Hardware Cost

Encoding Algorithm	NRZ-DBI	PAM4-DBI	PAM4-MF	PAM4-Sort
Word Length (bits)	8	16	16	16
Flag Wires (lanes)	1	1	1	2.5
Required Hardware Logic	2 Counters, Bitwise inverter	4 Counters, Bitwise inverter	4 Counters, Symbol swapper	4 Counters, Symbol sorter, Symbol swapper

Experimental Setup

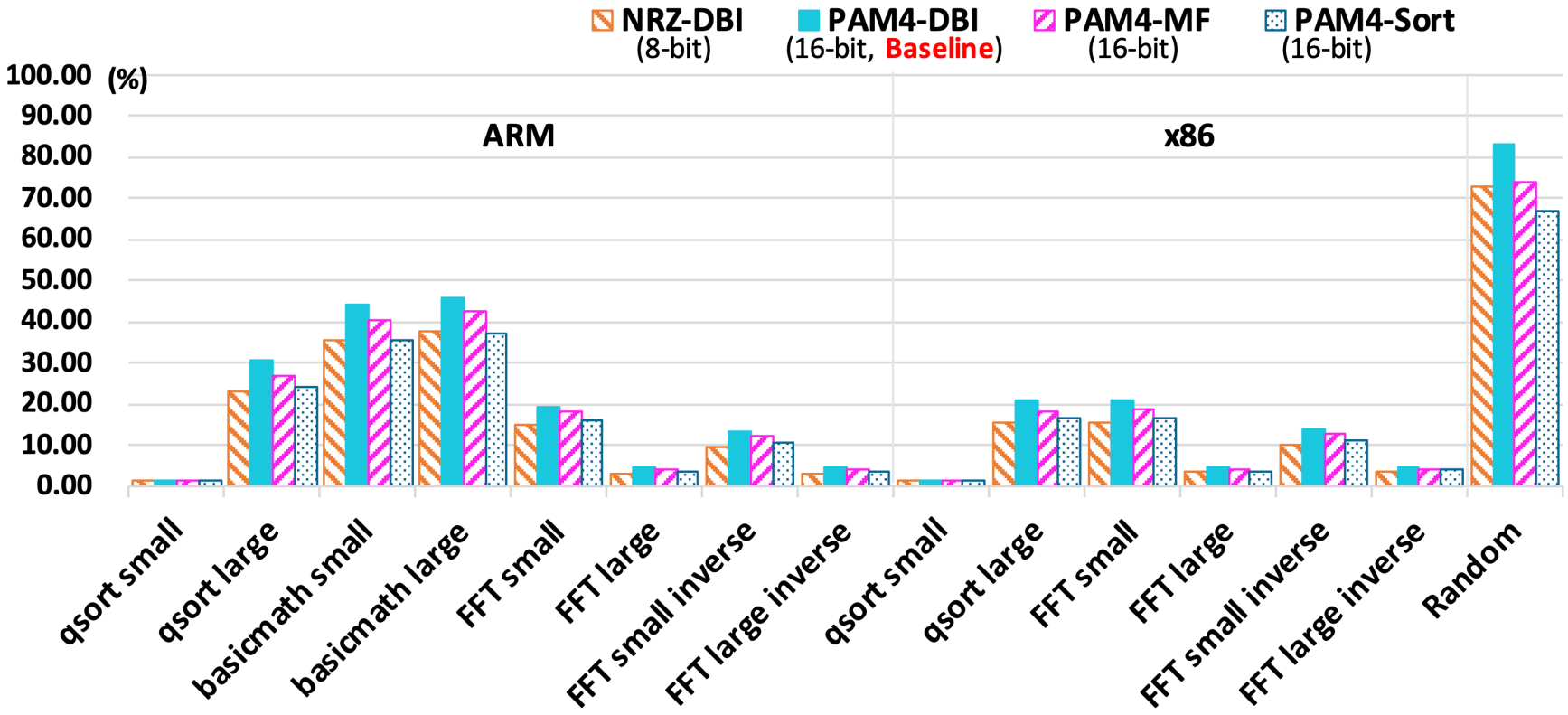


Processor	Out of order (O3)
Cache	L1: I-cache (32kB) D-cache (64kB) L2 (2MB)
DRAM	DDR4_2400_16x4, 1 channel, 512MB

$$Ratio_{power}(\%) = \frac{Power_{DBE}}{Power_{nonDBE}} \times 100(\%)$$

Termination Power

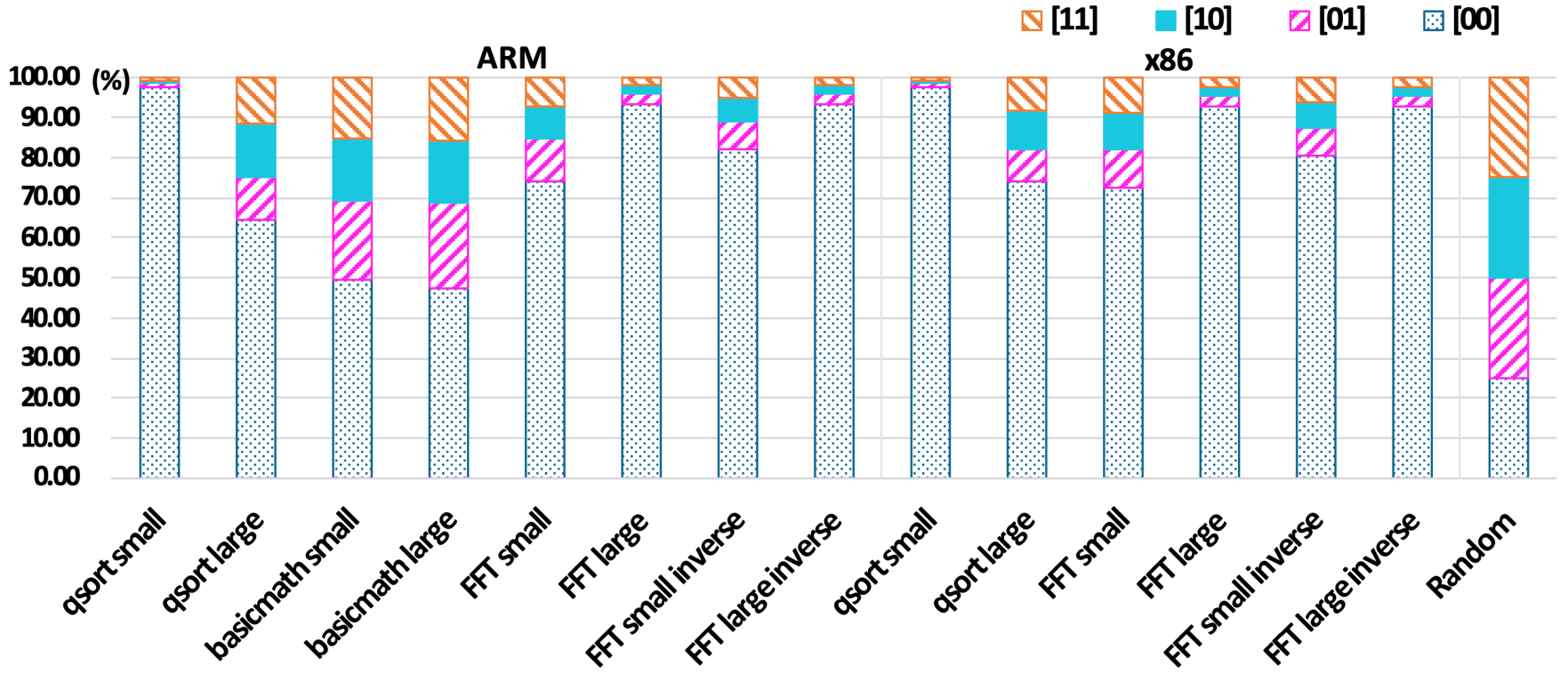
- Termination Power is reduced in **ALL cases**
- Overall effectiveness of the encoding schemes for POD I/Os



Termination Power

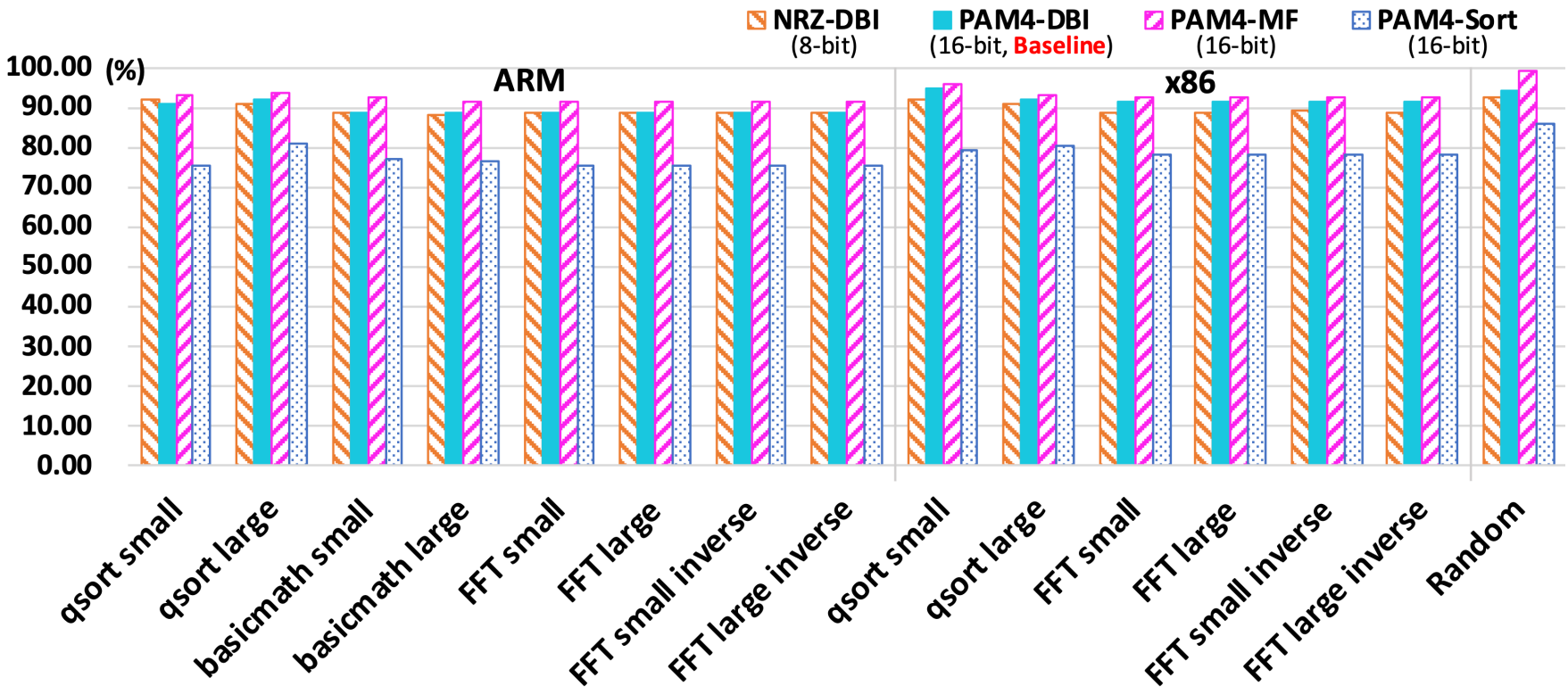
- Relations between **power saving ratio** and **frequency of 00 symbols**

⇨ The more **00**, the more **effective** it is



Switching Power

- Switching power was reduced for all benchmark programs
  - PAM4-MF by 7-9% and PAM4- Sort by more than 20%



Summary

PAM4-MF	• **Swaps** the **most frequent** symbol with the least power-consuming symbol **11** • **Decent improvement** (26.15% ~ 98.06%) compared with PAM4-DBI (17.02% ~ 98.46%) • **Simple** procedure
PAM4-Sort	• **Sorts** the entire symbol constellation based on their **frequencies** and **power consumption** • **Highest energy efficiency** (32.93% ~ 98.77)

- Both reduce **termination power** and **dynamic power**
- **Future work**
  - **Physical implementation** to study their feasibility at the circuit level

HYU

Thank You!

ICCD 2022 in Lake Tahoe, CA

Dept. of Electronic Engineering, Hanyang University
IoT Lab & Nifty Chips Lab