2017/01/21
第65回システムLSI合同ゼミ＠早稲田大学西早稲田キャンパス

DA変換器の線形性向上に向けた魔方陣アルゴリズムの検討

群馬大学大学院
理工学府 理工学専攻
情報通信システム分野第二　小林研究室
修士2年　東野　将史,
Shaiful Nizam Bin Mohyar,　小林　春夫

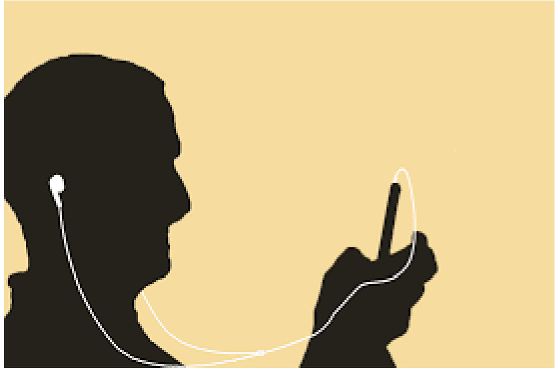
OUTLINE

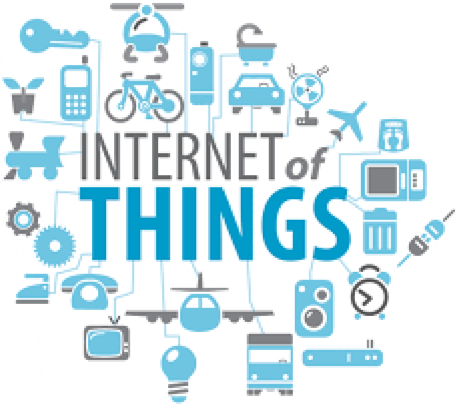
- Research Objective
- Current Steering DAC
- What is Magic Square ?
- Proposed Algorithm
- Simulation Results
- Conclusion

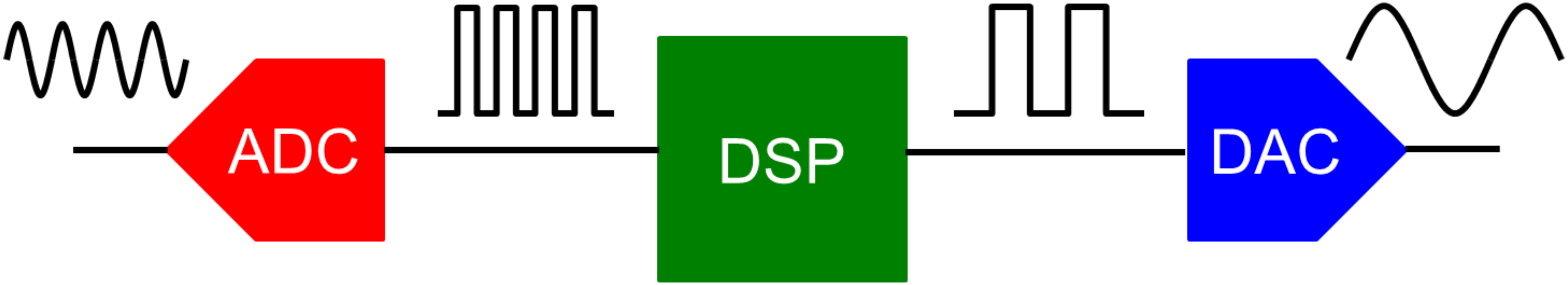
OUTLINE

- Research Objective
- Current Steering DAC
- What is Magic Square ?
- Proposed Algorithm
- Simulation Results
- Conclusion

研究背景







高性能なDA変換器が求められている

研究目的

古典数学を用いた線形性向上アルゴリズムの考案

■ DA変換器の課題

DA変換器の入出力関係は、理想的に線形関係 ☺

↓

MOSFETのミスマッチにより、入出力関係が非線形 ☹

↓

魔方陣を用いたキャリブレーション・アルゴリズム

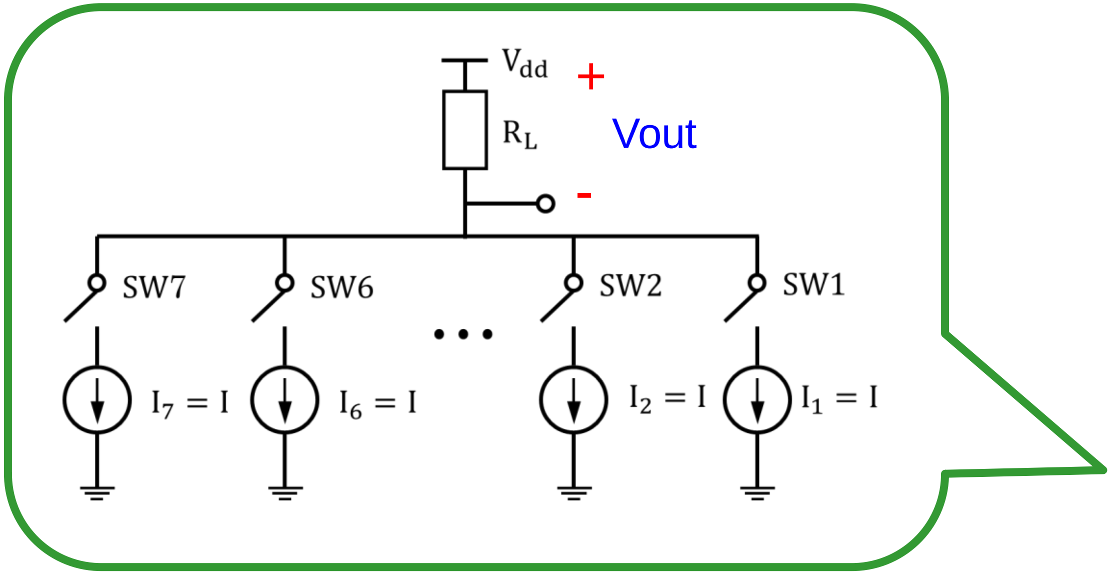
OUTLINE

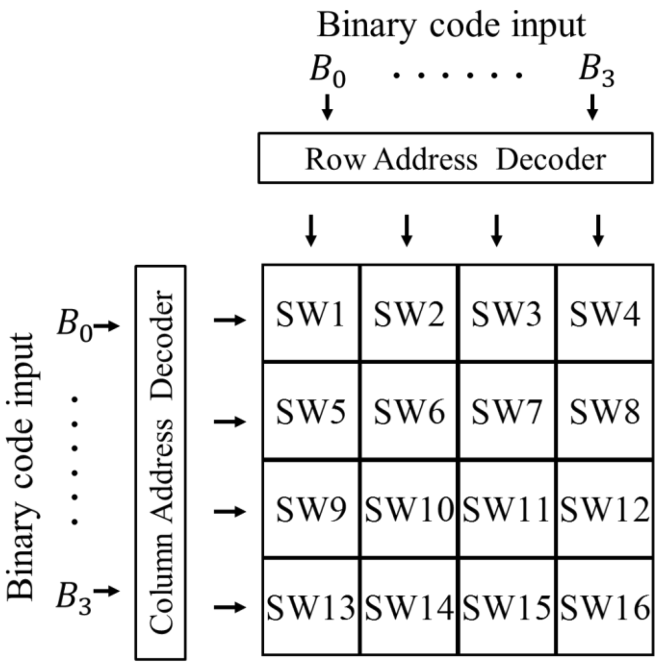
- Research Objective
- Current Steering DAC
- What is Magic Square ?
- Proposed Algorithm
- Simulation Results
- Conclusion

Circuit and Features of Unary Current-Steering DAC

- Identical current sources
- Small glitch
- Inherent monotonicity
- High speed ☺

- Large circuits
  - Decoder
  - Many switches and current sources ☹





Binary code input

B_0 . . . . B_3

Row Address Decoder

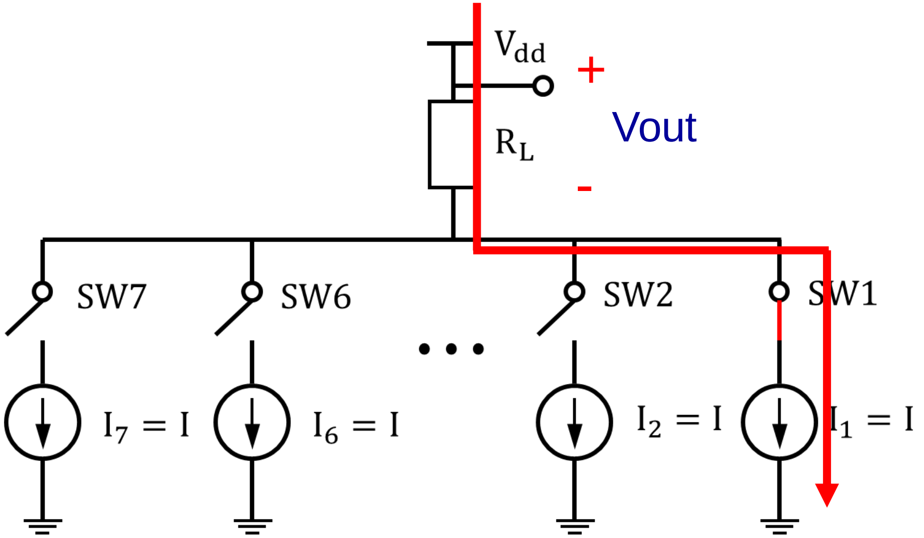
Binary code input

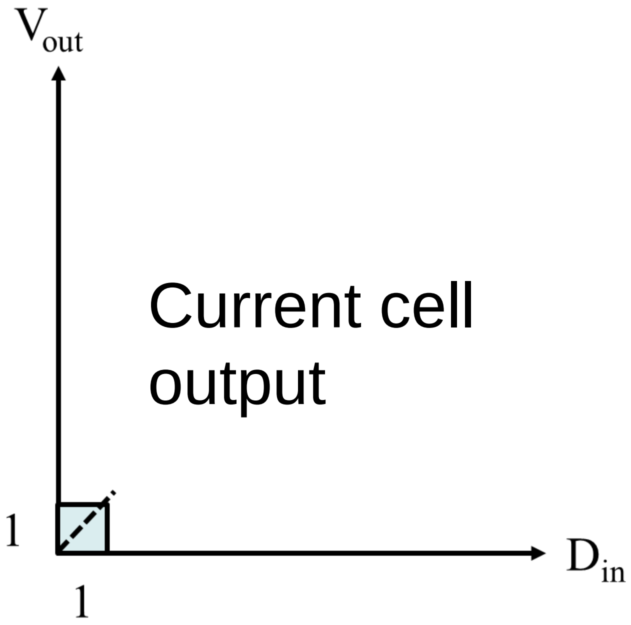
B_0 . . . . . . B_3

Column Address Decoder

SW1	SW2	SW3	SW4
SW5	SW6	SW7	SW8
SW9	SW10	SW11	SW12
SW13	SW14	SW15	SW16

Operation of Current Steering DAC (1)



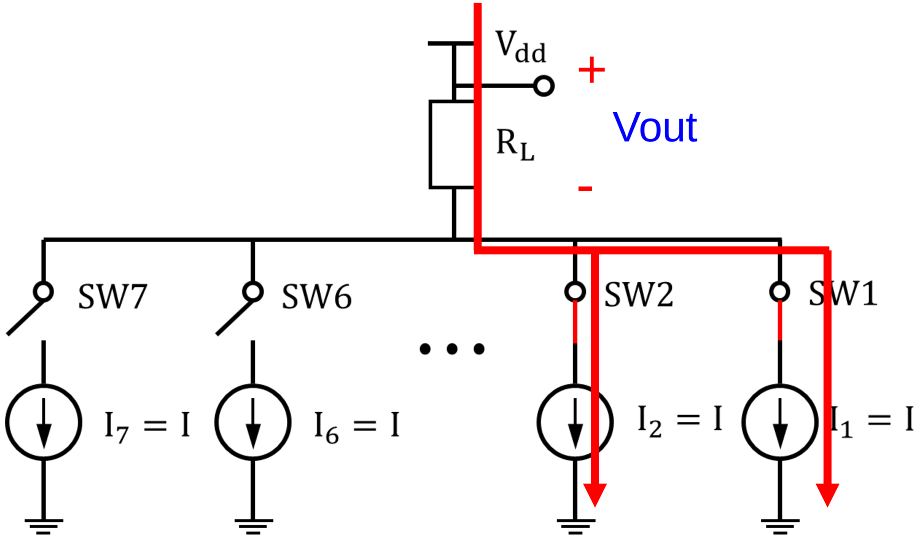


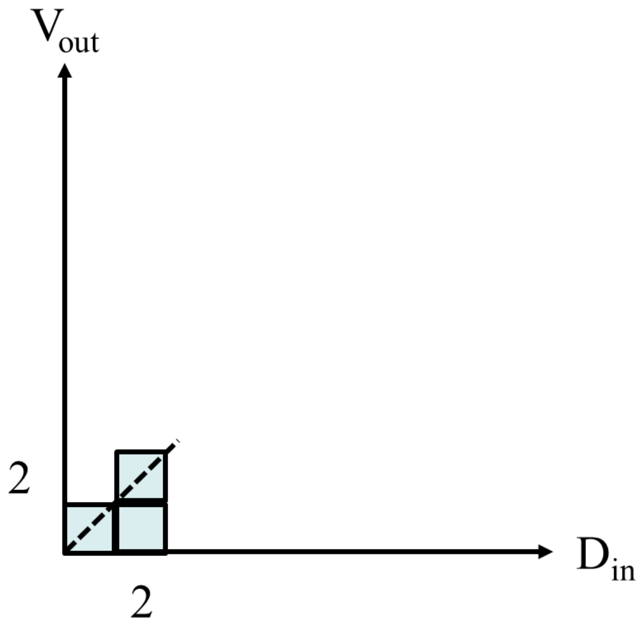
Current cell output

Digital input = 1 , 1 current source.

$$V_{out} = R_L I_1$$

Operation of Current Steering DAC (2)



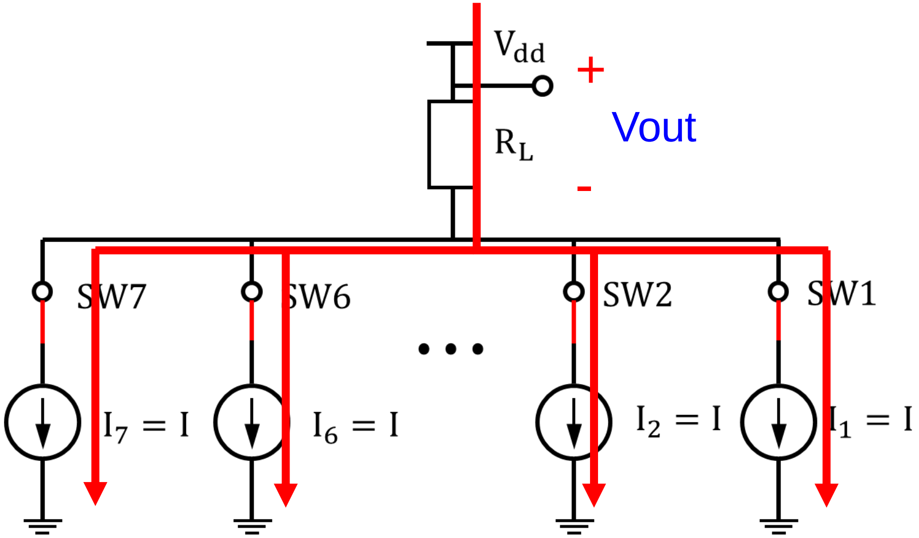


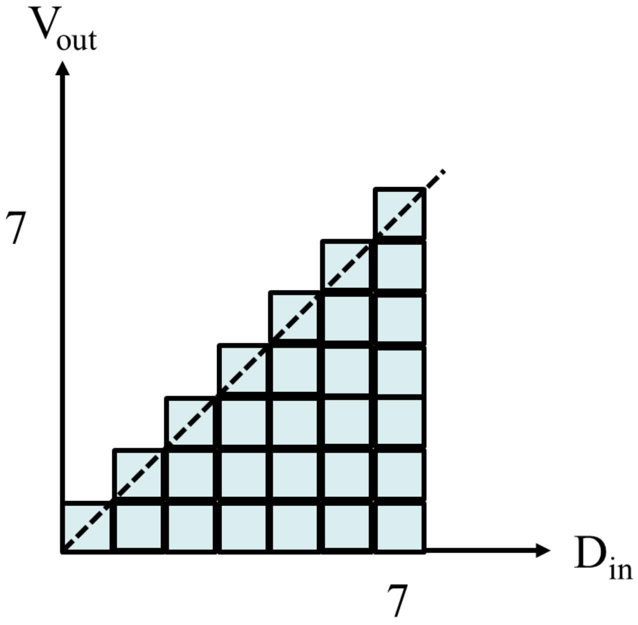
Digital input = 1 , 1 current source.

“ = 2 , 2 current sources.

$$V_{out} = R_L(I_1+I_2)$$

Operation of Current Steering DAC (3)





Digital input = 1 , 1 current source.
“ = 2 , 2 current sources.

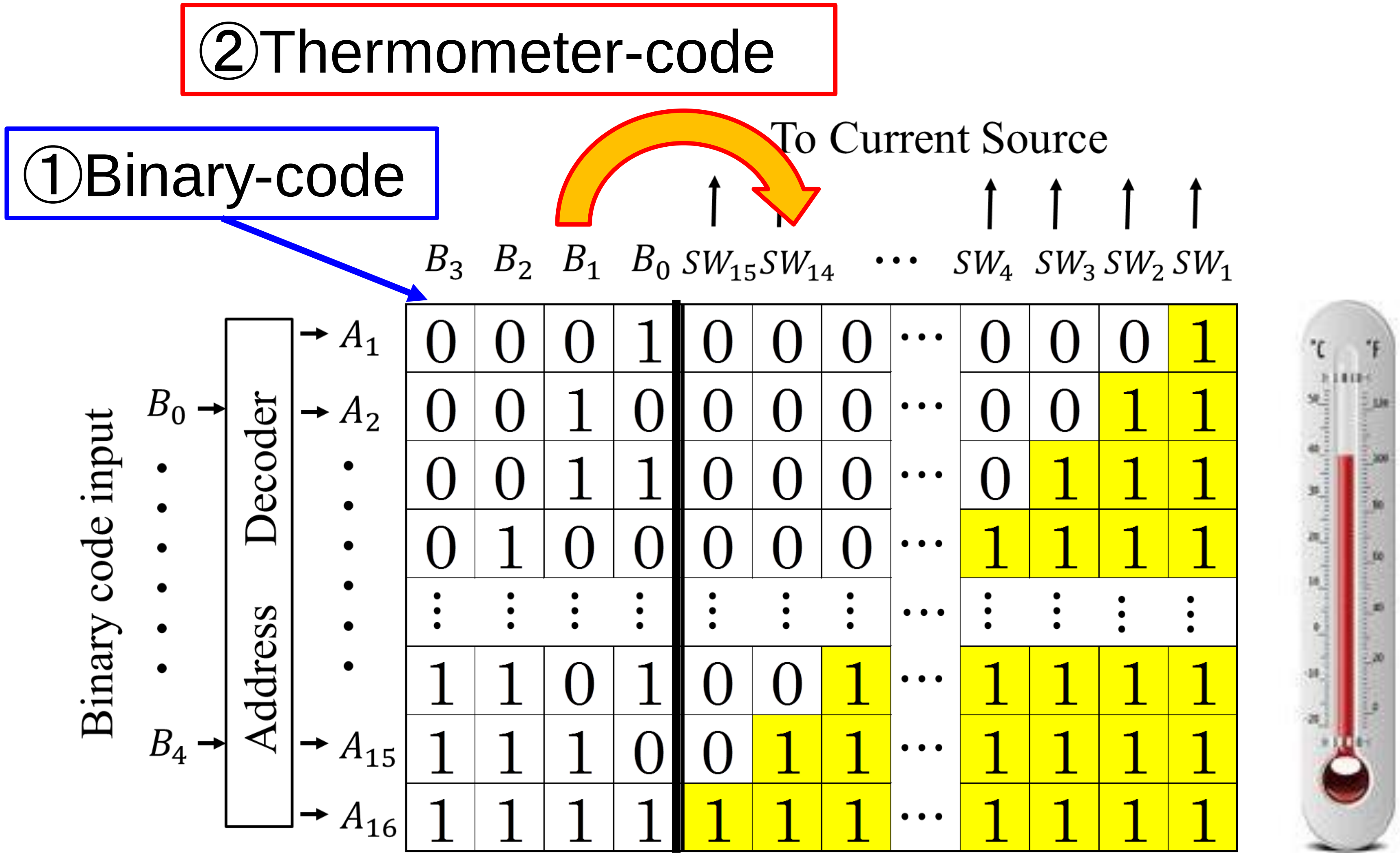
“ = 7 , 7 current sources.

$$V_{out} = R_L(I_1+I_2+\cdots+I_7)$$

$$I_1 = I_2 = \cdots = I_6 = I_7$$

DAC is perfectly linear

Conventional Unary DAC Decoder



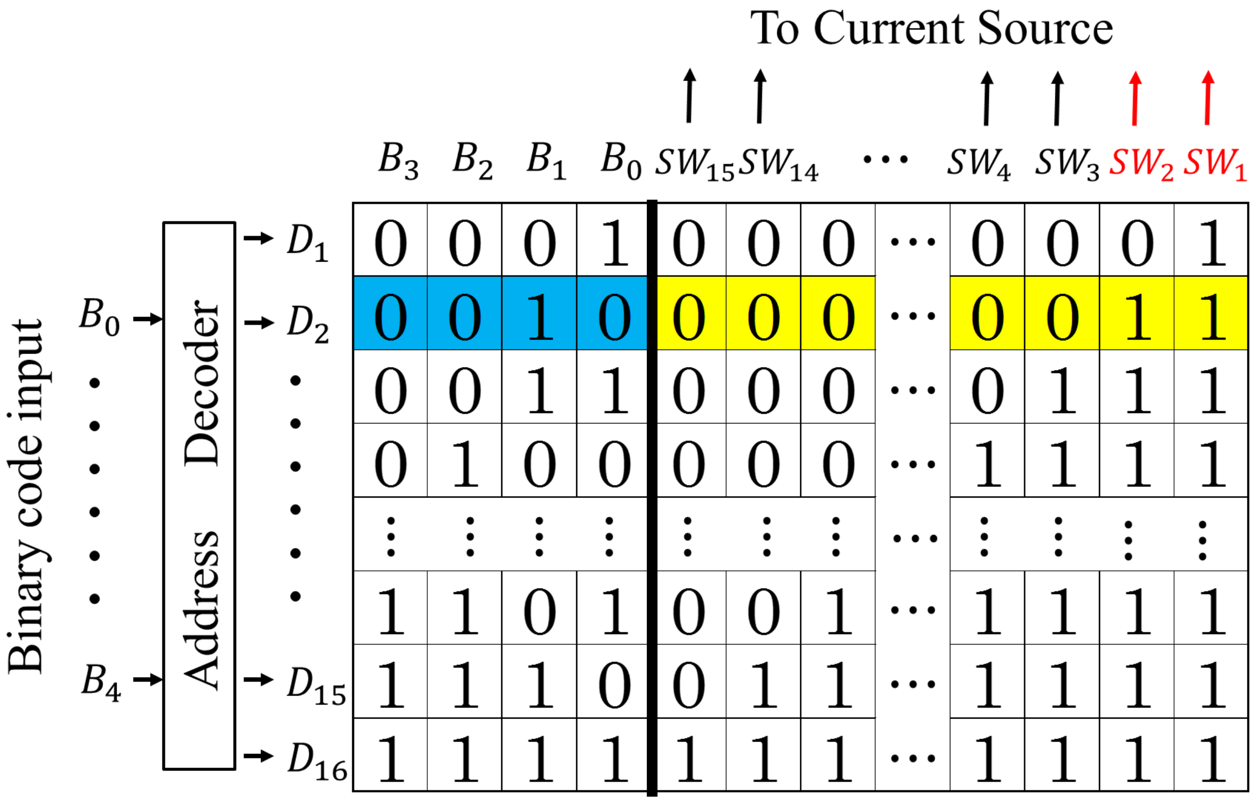
②Thermometer-code

①Binary-code

	B_3	B_2	B_1	B_0	SW_{15}	SW_{14}		⋯	SW_4	SW_3	SW_2	SW_1
A_1	0	0	0	1	0	0	0	⋯	0	0	0	1
A_2	0	0	1	0	0	0	0	⋯	0	0	1	1
⋮	0	0	1	1	0	0	0	⋯	0	1	1	1
	0	1	0	0	0	0	0	⋯	1	1	1	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋯	⋮	⋮	⋮	⋮
	1	1	0	1	0	0	1	⋯	1	1	1	1
A_{15}	1	1	1	0	0	1	1	⋯	1	1	1	1
A_{16}	1	1	1	1	1	1	1	⋯	1	1	1	1

Operation of Unary DAC Decoder Example 1

- Digital binary input (0010)
  ➡ Thermometer code (0000 0000 0000 0011)
  ➡ 2 current cells turn on.



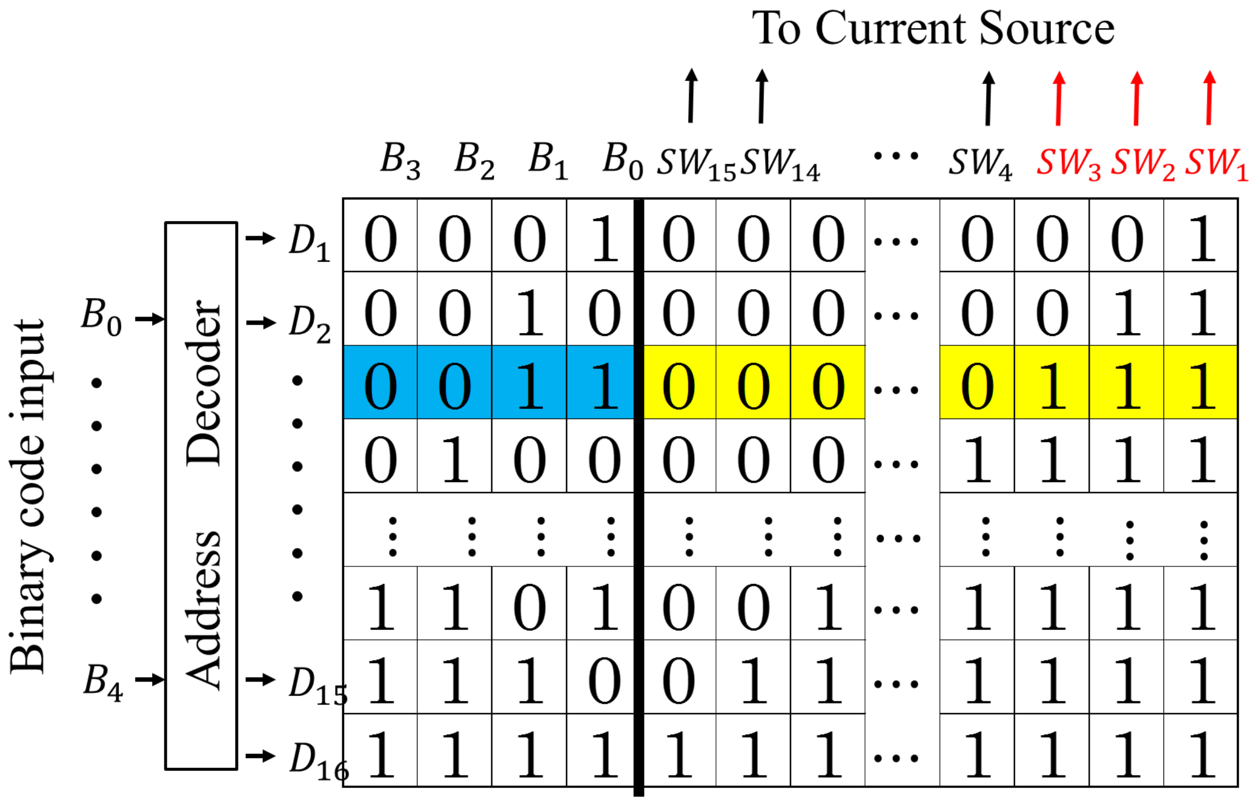
	B_3	B_2	B_1	B_0	SW_{15}	SW_{14}		⋯	SW_4	SW_3	SW_2	SW_1
D_1	0	0	0	1	0	0	0	⋯	0	0	0	1
D_2	0	0	1	0	0	0	0	⋯	0	0	1	1
	0	0	1	1	0	0	0	⋯	0	1	1	1
	0	1	0	0	0	0	0	⋯	1	1	1	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋯	⋮	⋮	⋮	⋮
	1	1	0	1	0	0	1	⋯	1	1	1	1
D_{15}	1	1	1	0	0	1	1	⋯	1	1	1	1
D_{16}	1	1	1	1	1	1	1	⋯	1	1	1	1

Operation of Unary DAC Decoder
Example 2

- Digital binary input (0011)

➡ Thermometer code (0000 0000 0000 0111)

➡ 3 current cells turn on.



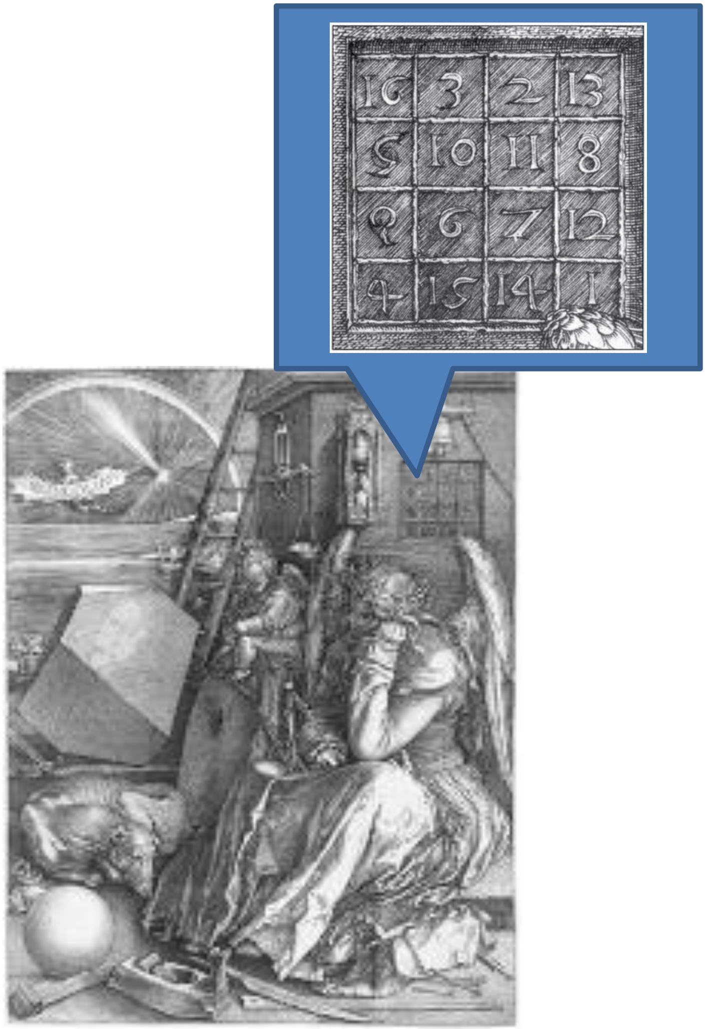
	B_3	B_2	B_1	B_0	SW_{15}	SW_{14}		⋯	SW_4	SW_3	SW_2	SW_1
D_1	0	0	0	1	0	0	0	⋯	0	0	0	1
D_2	0	0	1	0	0	0	0	⋯	0	0	1	1
	0	0	1	1	0	0	0	⋯	0	1	1	1
	0	1	0	0	0	0	0	⋯	1	1	1	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋯	⋮	⋮	⋮	⋮
	1	1	0	1	0	0	1	⋯	1	1	1	1
D_{15}	1	1	1	0	0	1	1	⋯	1	1	1	1
D_{16}	1	1	1	1	1	1	1	⋯	1	1	1	1

OUTLINE

- Research Objective
- Current Steering DAC
- What is Magic Square ?
- Proposed Algorithm
- Simulation Results
- Conclusion

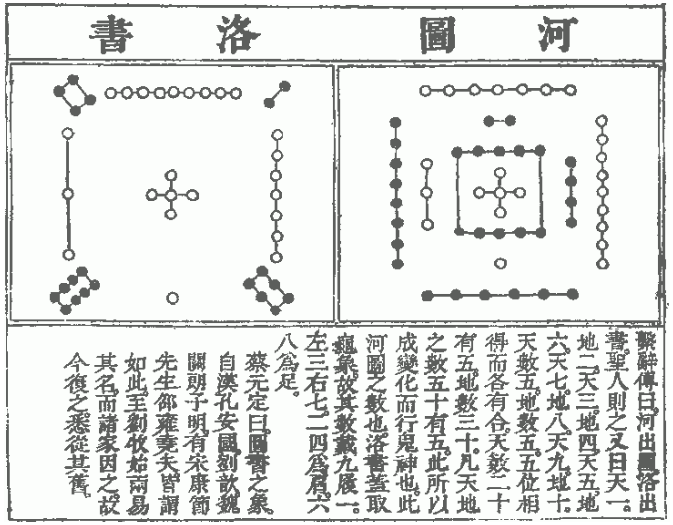
What is Magic Square(魔方陣) ?

みなさん、“魔方陣”をご存知でしょうか。



What is Magic Square ?

- Classical mathematics
- Origin from Chinese academia
- “Constant sum” characteristics
- Varieties of magic squares



河圖

洛書

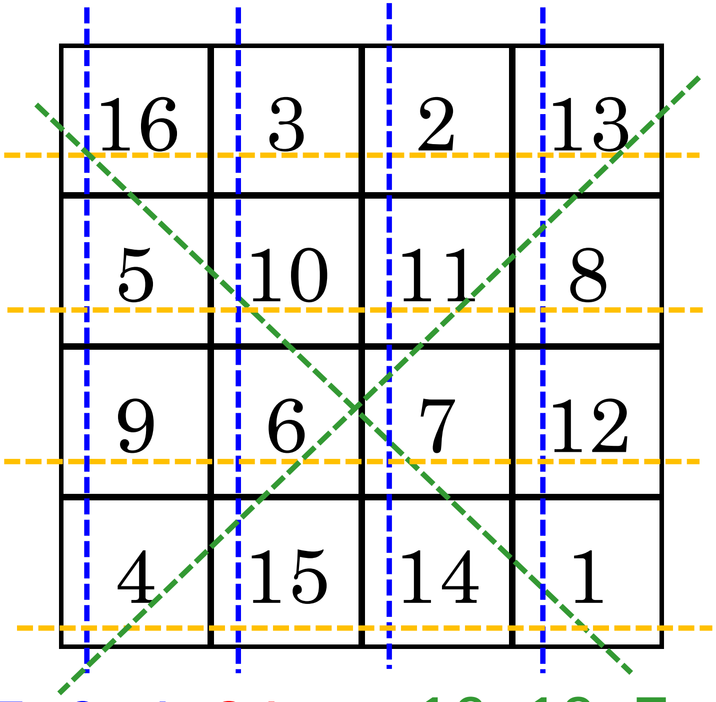
繫辭傳曰。河出圖洛出書。聖人則之。又曰天一。地二。天三。地四。天五。地六。天七。地八。天九。地十。天數五。地數五。五位相得而各有合。天數二十有五。地數三十。凡天地之數五十有五。此所以成變化而行鬼神也。此河圖之數也。洛書蓋取龜象。故其數戴九履一。左三右七。二四為肩。六八為足。

蔡元定曰。圖書之象。自漢孔安國劉歆魏關朗子明有宋康節先生邵雍堯夫皆謂如此。至劉牧始兩易其名。而諸家因之。故今復之。悉從其舊。

3x3 魔方陣

4	9	2
3	5	7
8	1	6

Features of Magic Square



16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

16+3+2+13=34

16+5+9+4=34

16+10+7+1=34

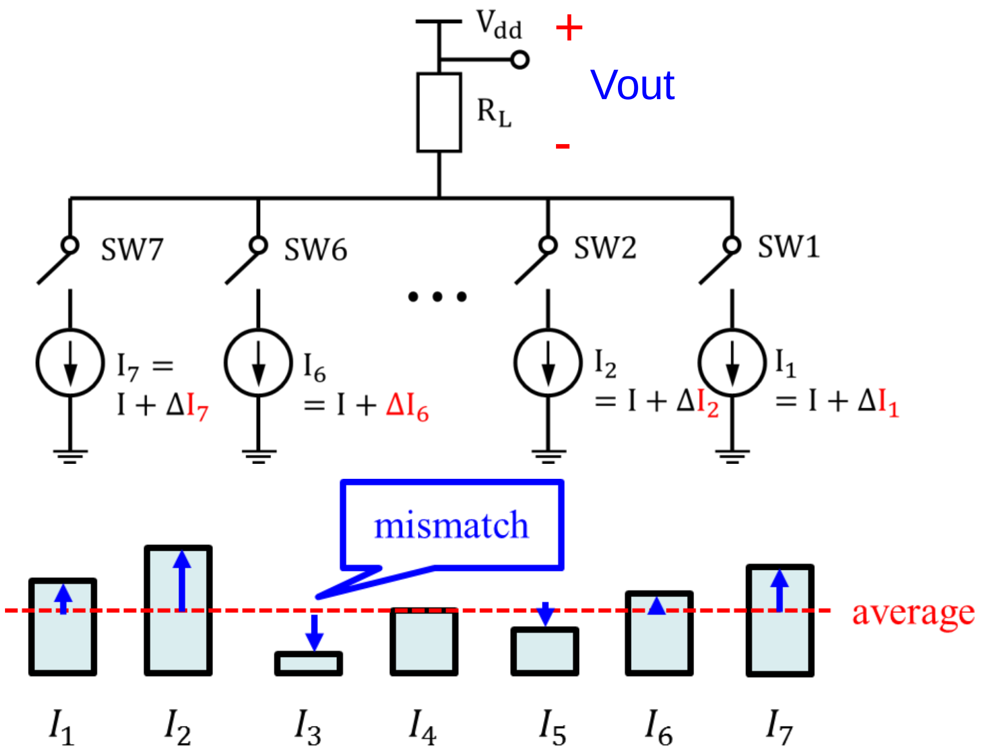
- Constant Sum

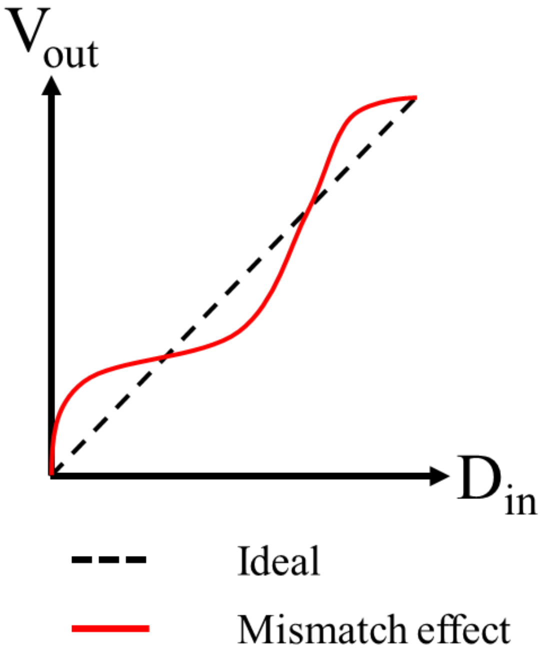
  - Row, column, diagonal → 魔方陣 is Good balance

OUTLINE

- Research Objective
- Current Steering DAC
- What is Magic Square ?
- **Proposed Algorithm**
- Simulation Results
- Conclusion

Unit Current Source Mismatch Problem



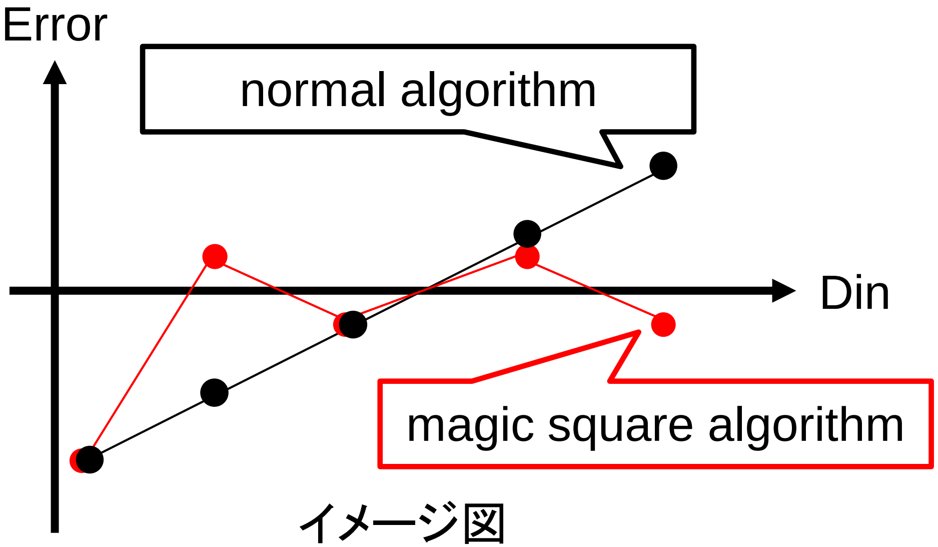


In practice, current sources have mismatches.

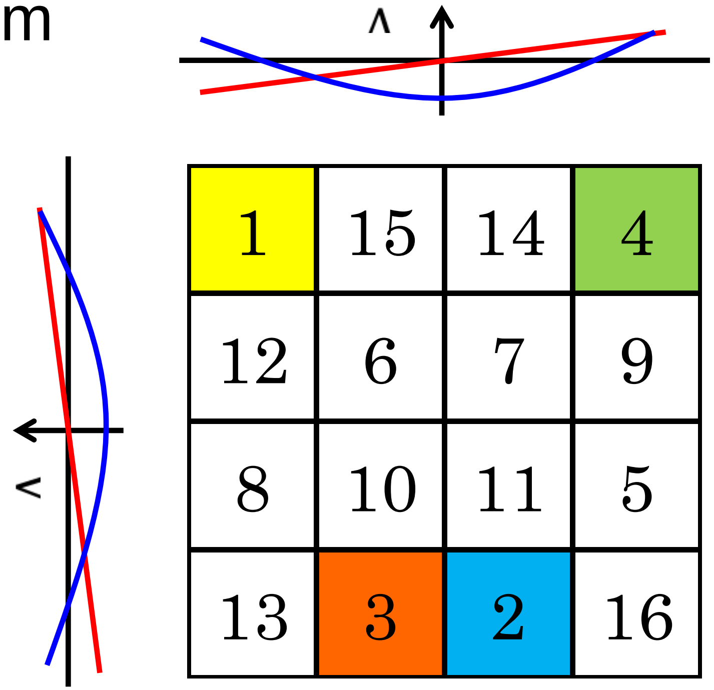
➡ DAC becomes non-linear.

Possibility of Using Magic Square

- Semiconductor devices have random and systematic mismatches
- Changing the switching order ➡ Cancellation of mismatch effects
- We propose magic square algorithm



イメージ図

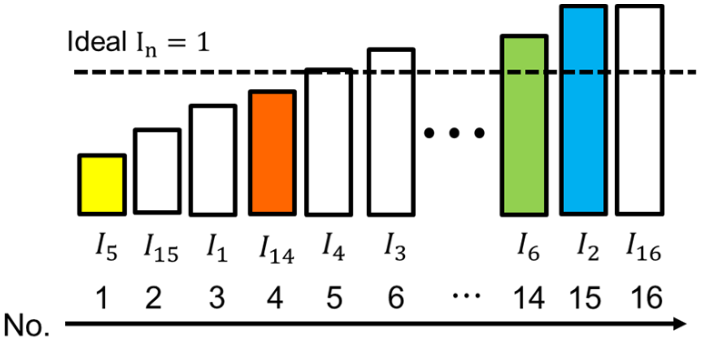


1	15	14	4
12	6	7	9
8	10	11	5
13	3	2	16

Inspired New Algorithm

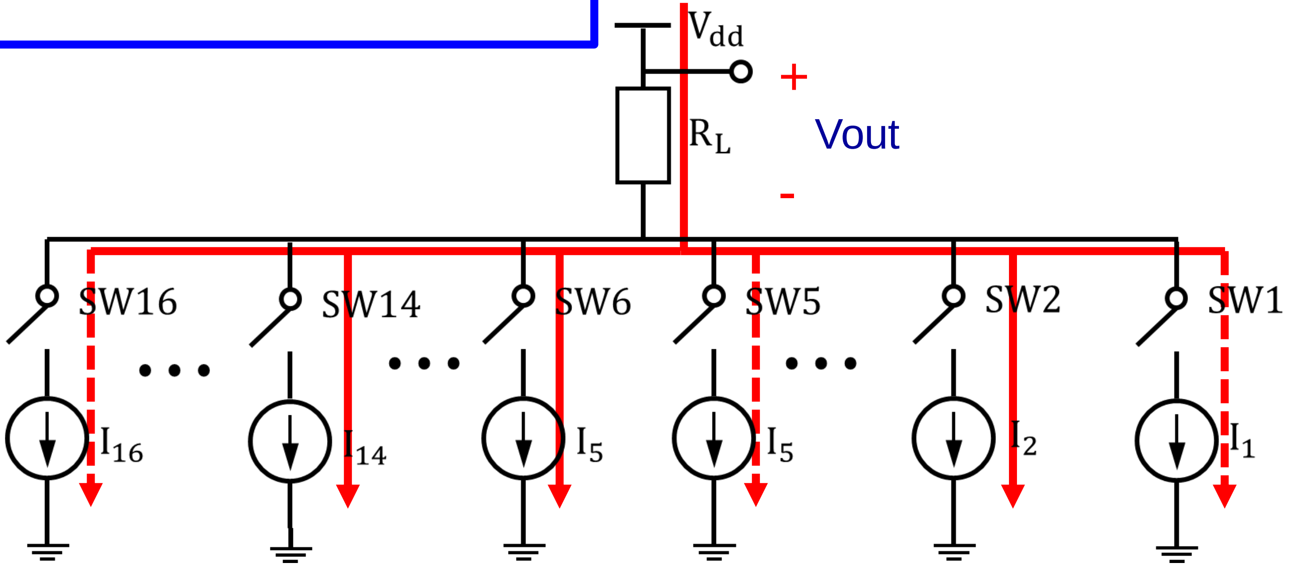
- Unit current source selection-order change algorithm
  - Mismatch effect cancellation

I_1	I_2	I_3	I_4
I_5	I_6	I_7	I_8
I_9	I_{10}	I_{11}	I_{12}
I_{13}	I_{14}	I_{15}	I_{16}

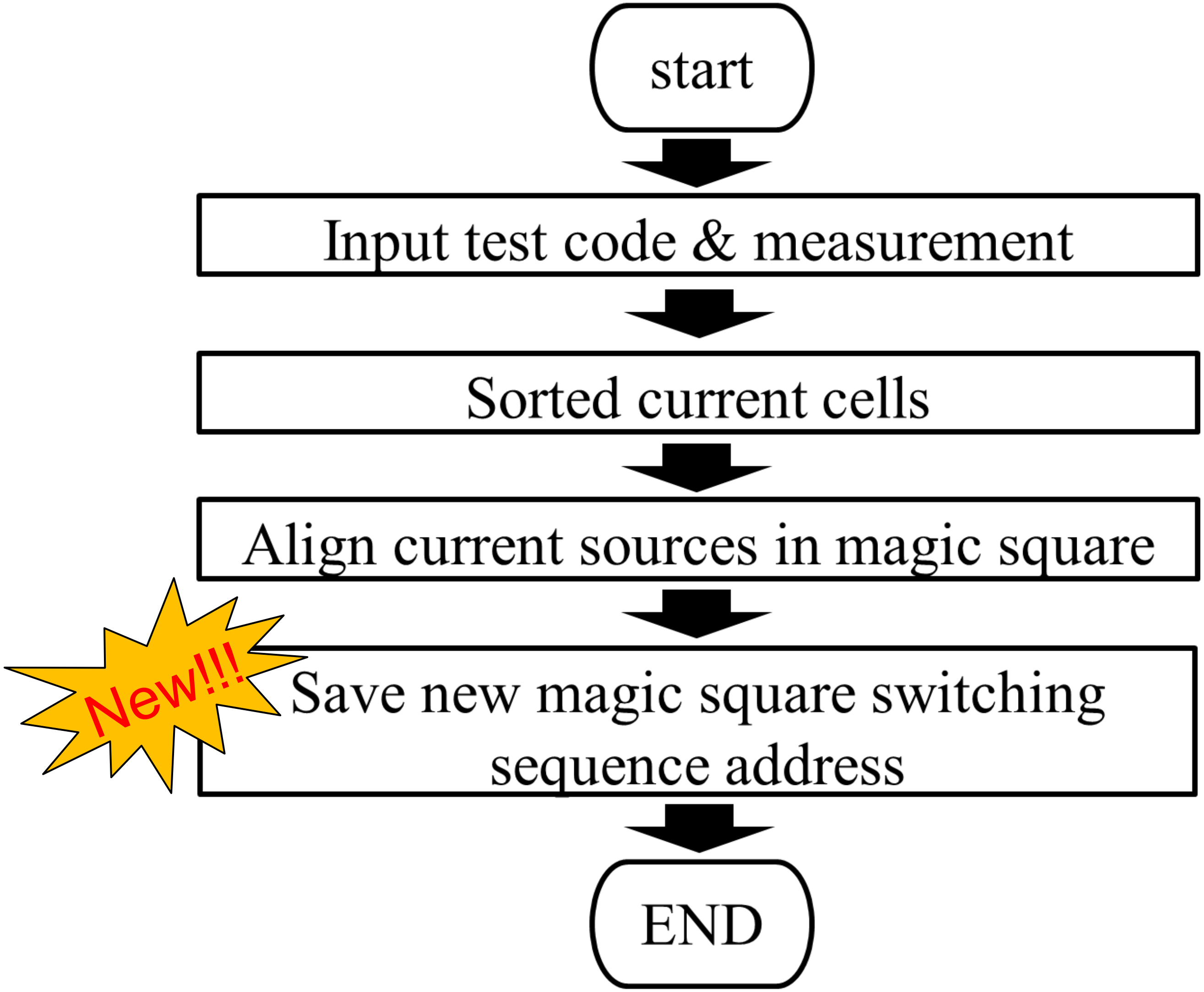


1. Measure the order of unit current cells
2. Align them virtually in magic square
3. Select current cells

1	15	14	4
12	6	7	9
8	10	11	5
13	3	2	16



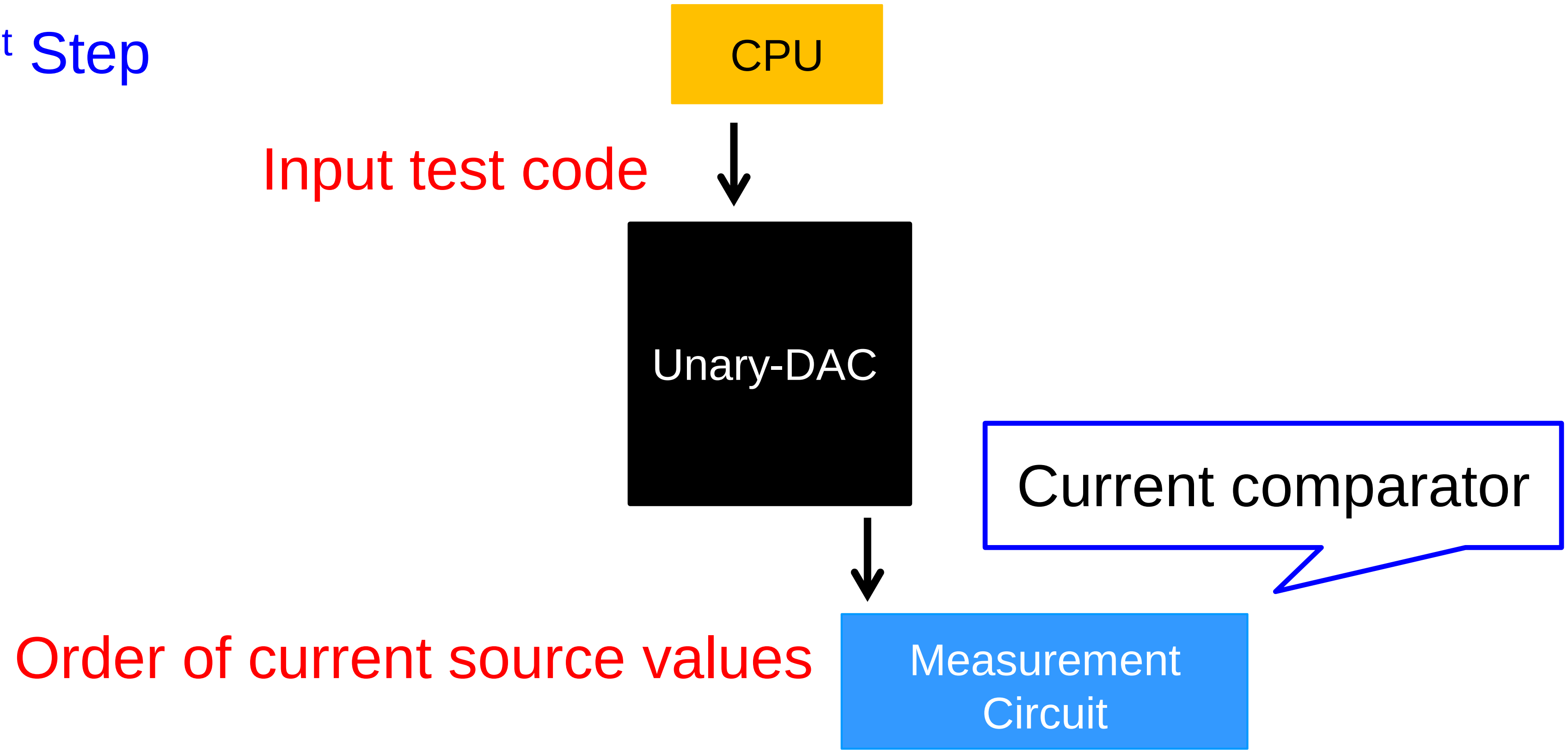
Proposed DAC Non-linearity Calibration Algorithm



start
Input test code & measurement
Sorted current cells
Align current sources in magic square
New!!!
Save new magic square switching sequence address
END

Proposed Algorithm
Input Test Code & Measurement

1st Step



CPU

Input test code

Unary-DAC

Current comparator

Order of current source values

Measurement Circuit

- CPU => input test code to unary-DAC cells
- Measurement circuit => order of current source values

Proposed Algorithm
Measure Order of Current Cells

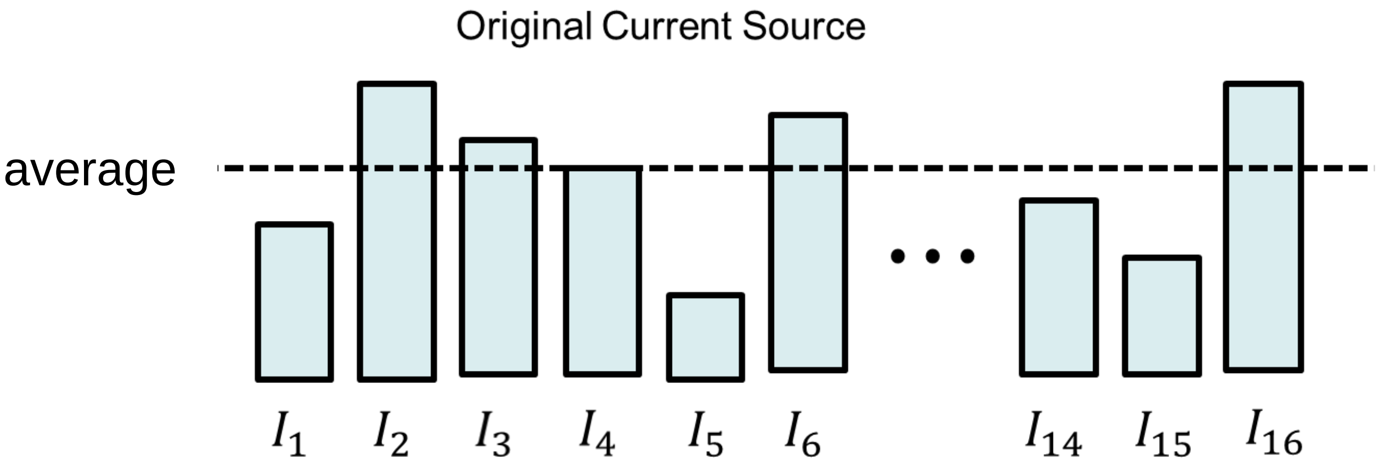
1st Step

- Measure the order of current cell values by a current comparator.
- Not need accurate value measurement.

4-bit case

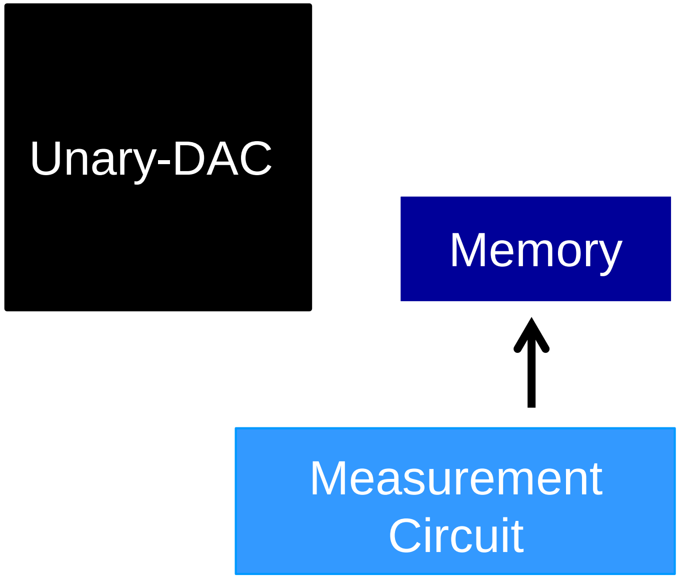
Current cell

I_1	I_2	I_3	I_4
I_5	I_6	I_7	I_8
I_9	I_{10}	I_{11}	I_{12}
I_{13}	I_{14}	I_{15}	I_{16}



Proposed Algorithm
Unit Current Source Sorting

2nd step



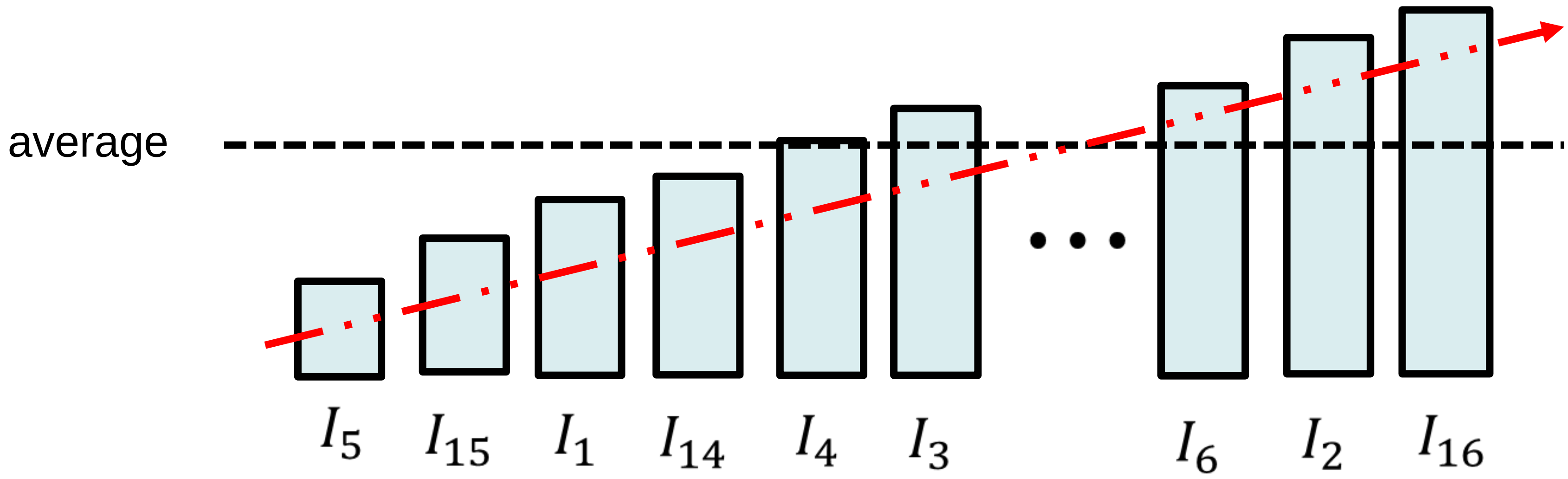
Sort and store the measured order of the unit current cell values into memory.

Proposed Algorithm
Unit Current Source Sorting

2nd step

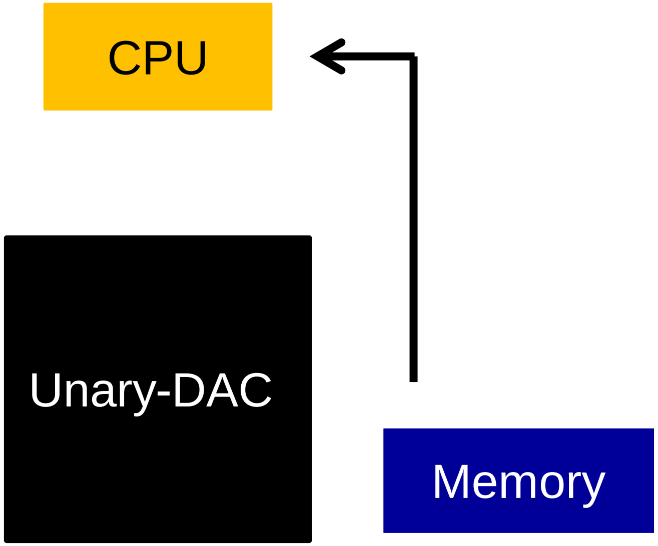
- Sort current source cells ascendingly.
- Store their information of cells number and value into memory.

Sorted Current Source



Proposed Algorithm
Current Source Sorting Based on Magic Square

3rd step



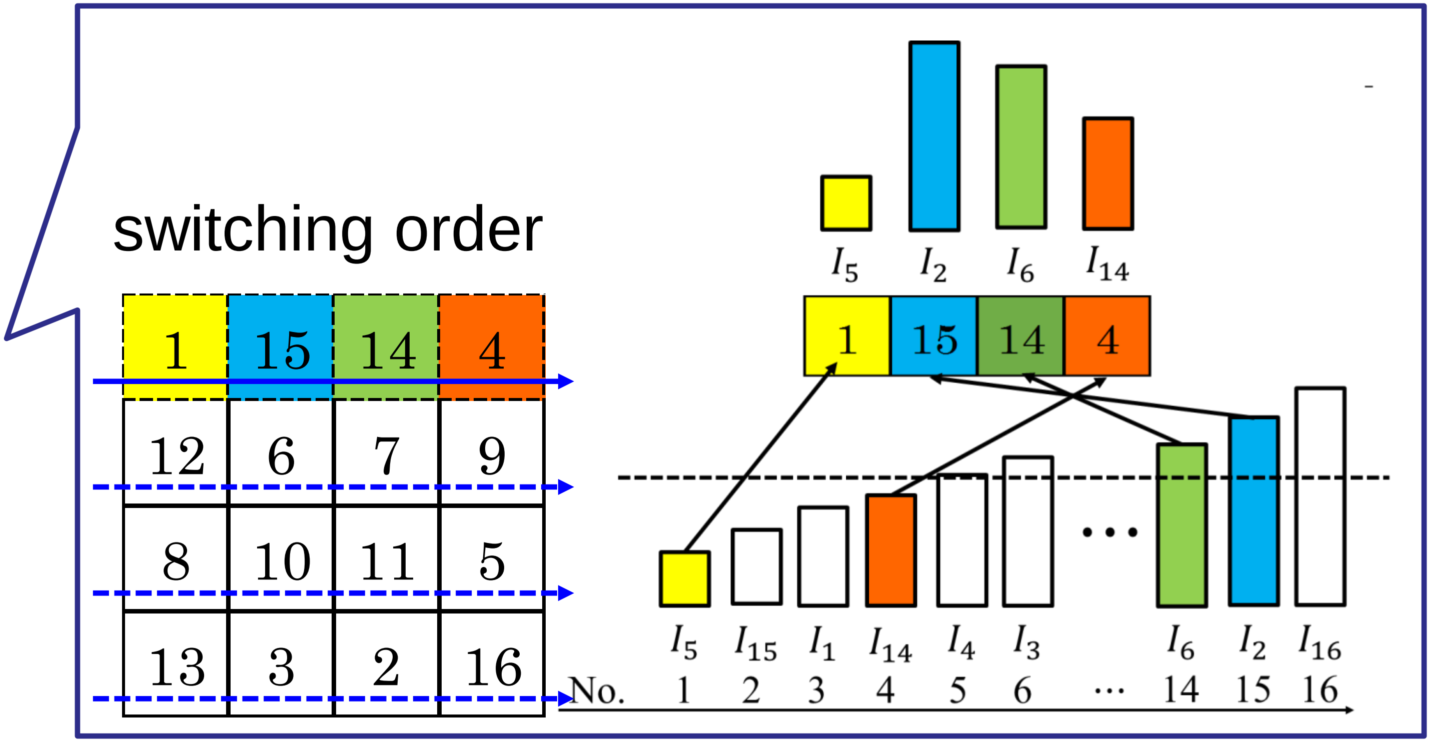
- Re-sort of current source values based on magic square

Proposed Algorithm
Current Source Sorting Based on Magic Square (1)

3rd step

- Re-sorted of current source values based on magic square
- Store its info in decoder look-up table

I_1	I_2	I_3	I_4
I_5	I_6	I_7	I_8
I_9	I_{10}	I_{11}	I_{12}
I_{13}	I_{14}	I_{15}	I_{16}



switching order

1	15	14	4
12	6	7	9
8	10	11	5
13	3	2	16

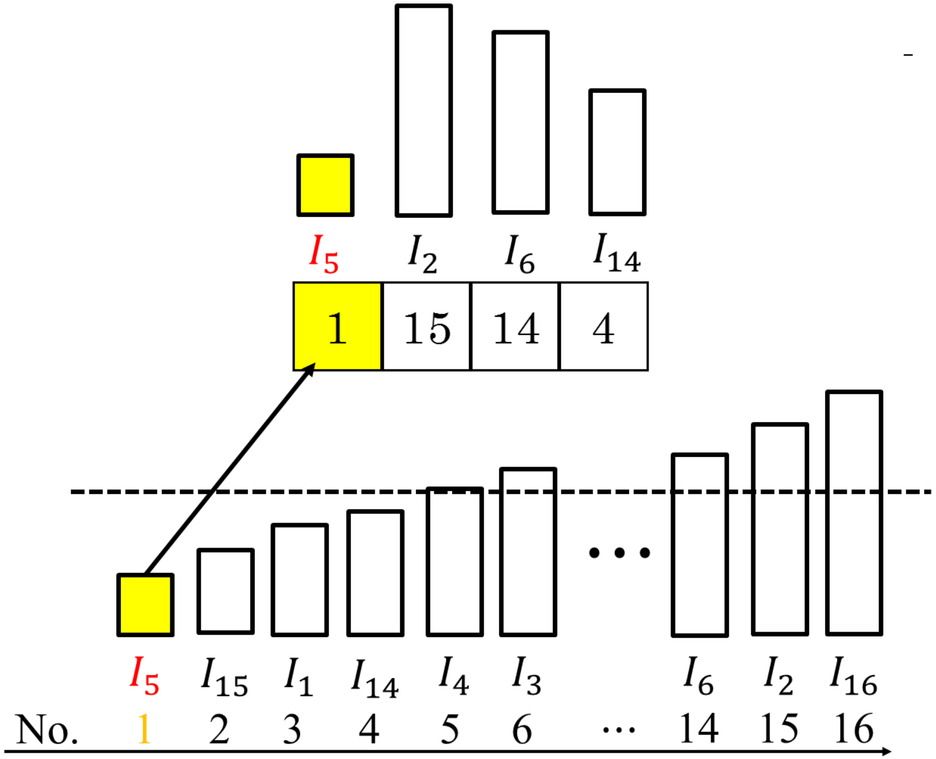
I_5 I_2 I_6 I_{14}
1 15 14 4
I_5 I_{15} I_1 I_{14} I_4 I_3 I_6 I_2 I_{16}
No. 1 2 3 4 5 6 … 14 15 16
…

Proposed Algorithm
Current Source Sorting Based on Magic Square (2)

- Digital binary input (0001)

  ➡ 1 current cells turn on

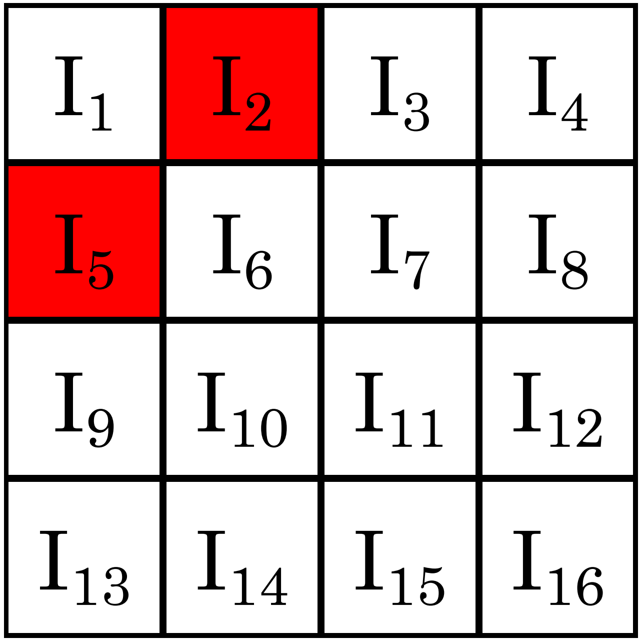
I_1	I_2	I_3	I_4
I_5	I_6	I_7	I_8
I_9	I_{10}	I_{11}	I_{12}
I_{13}	I_{14}	I_{15}	I_{16}



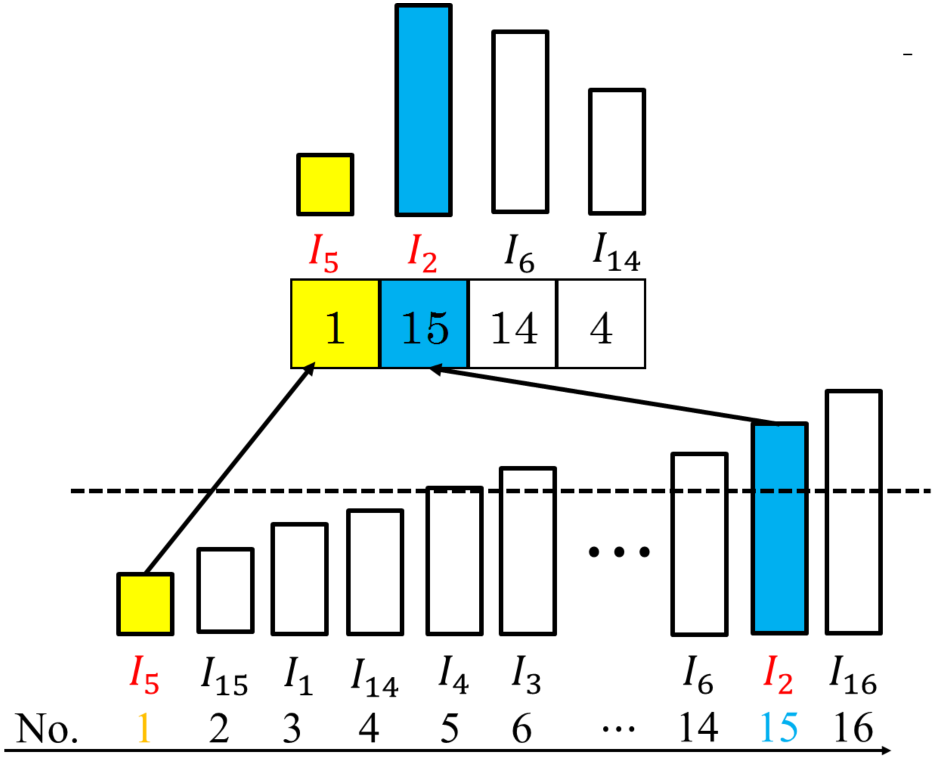
Proposed Algorithm
Current Source Sorting Based on Magic Square (3)

- Digital binary input (0010)

➡ 2 current cells turn on



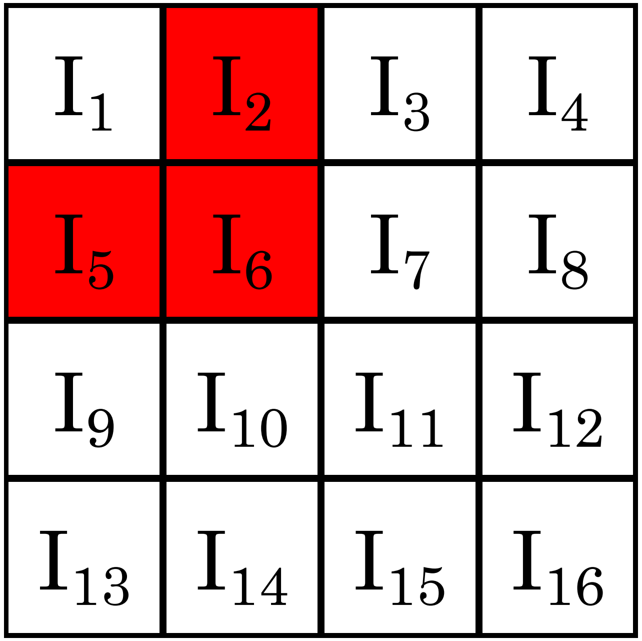
I_1	I_2	I_3	I_4
I_5	I_6	I_7	I_8
I_9	I_{10}	I_{11}	I_{12}
I_{13}	I_{14}	I_{15}	I_{16}



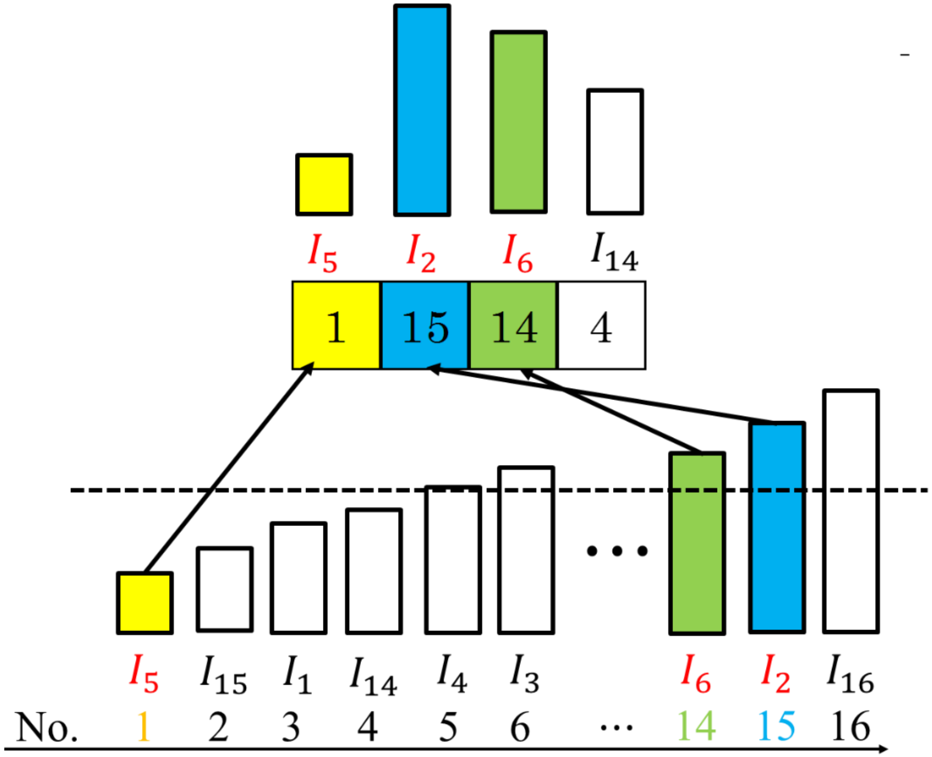
Proposed Algorithm
Current Source Sorting Based on Magic Square (4)

- Digital binary input (0011)

  ➡ 3 current cells turn on



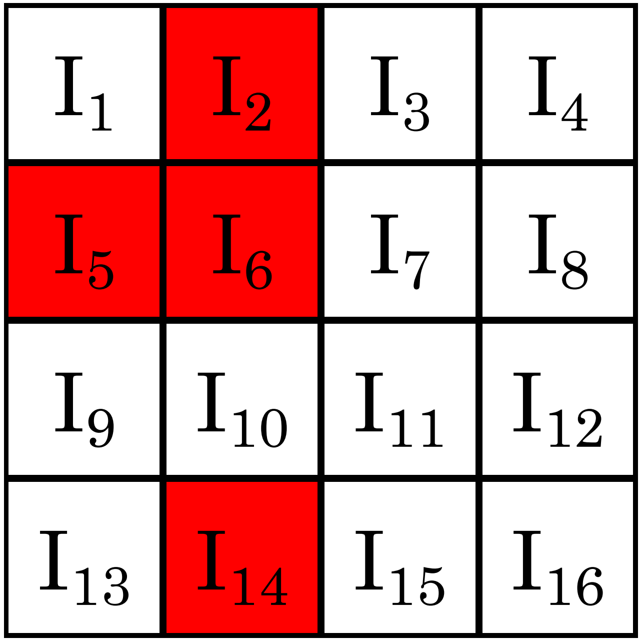
I_1	I_2	I_3	I_4
I_5	I_6	I_7	I_8
I_9	I_{10}	I_{11}	I_{12}
I_{13}	I_{14}	I_{15}	I_{16}



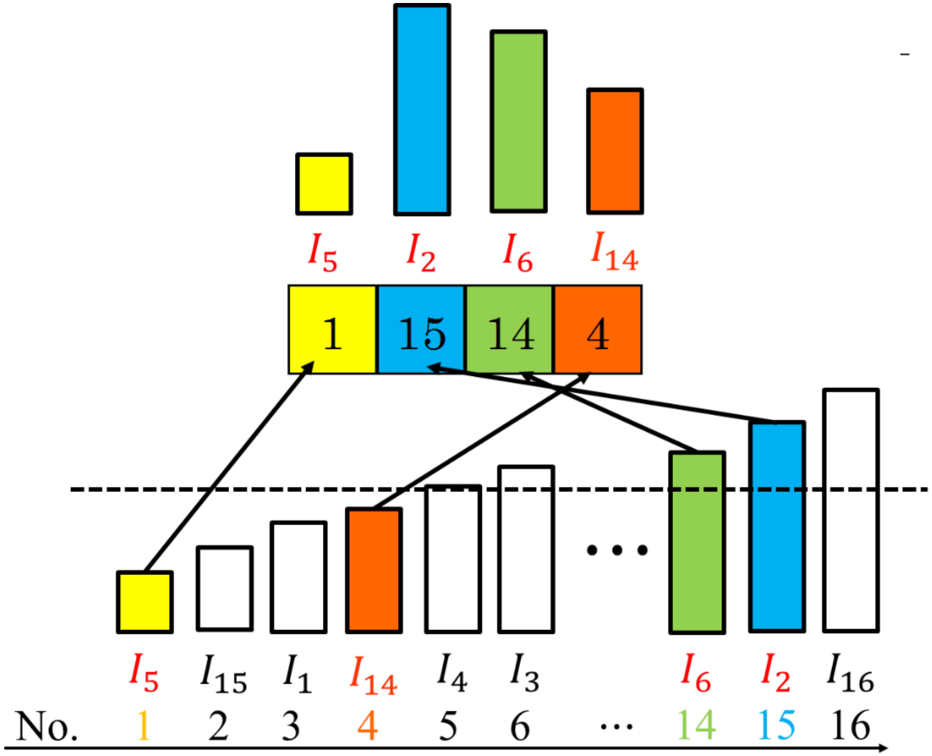
Proposed Algorithm
Current Source Sorting Based on Magic Square (5)

- Digital binary input (0100)

  ➡ 4 current cells turn on

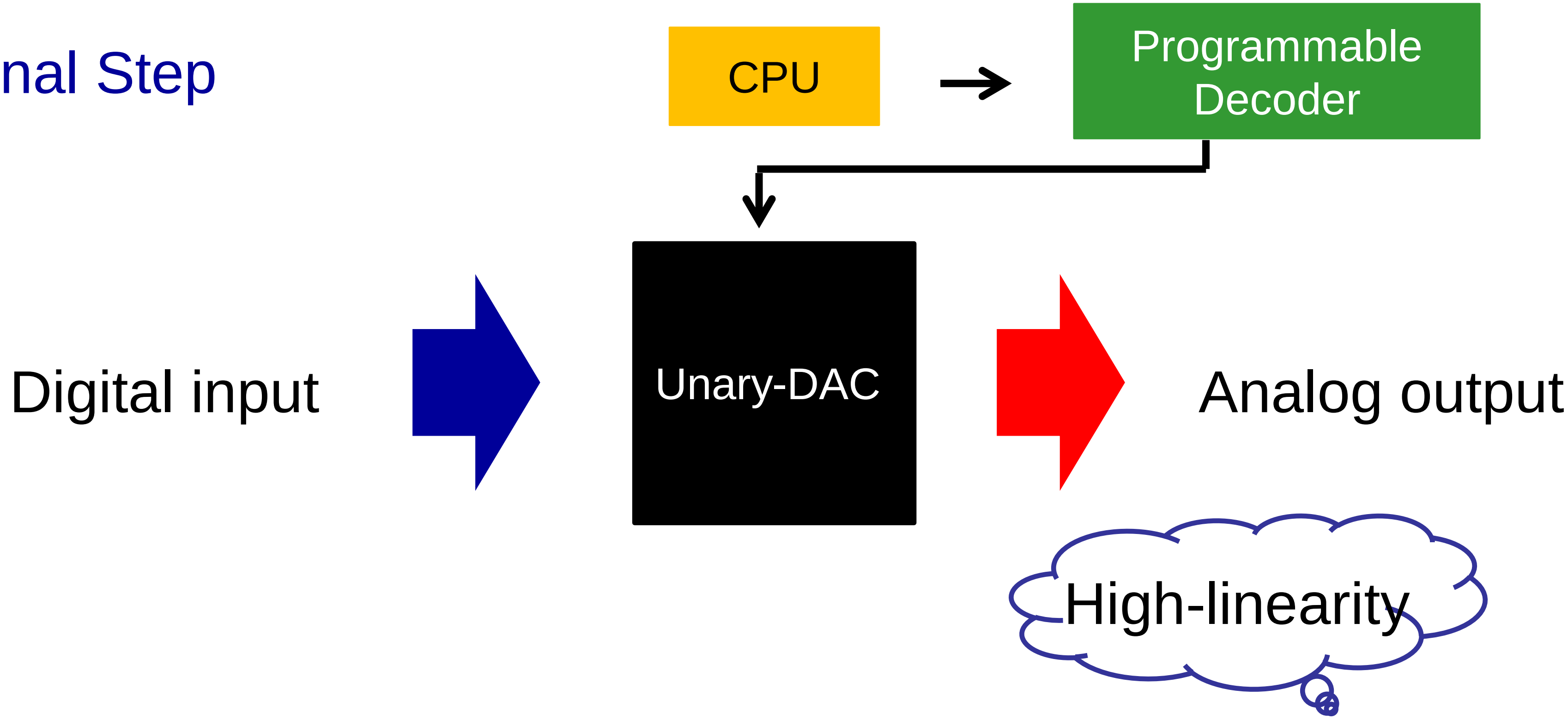


I_1	I_2	I_3	I_4
I_5	I_6	I_7	I_8
I_9	I_{10}	I_{11}	I_{12}
I_{13}	I_{14}	I_{15}	I_{16}



Proposed Algorithm LUT-Magic Square Decoder

Final Step



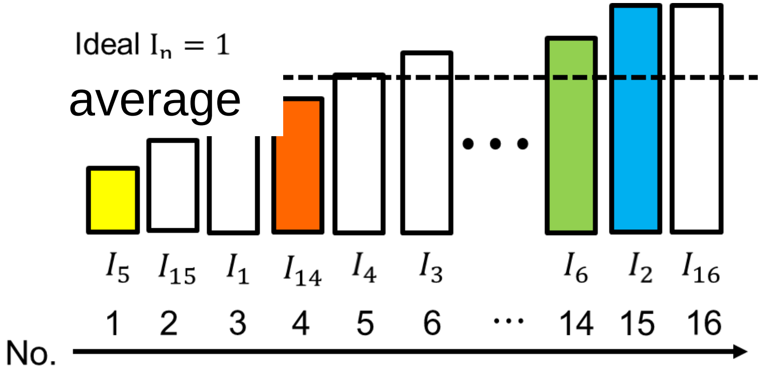
Store switching sequence based on magic square into programmable decoder.

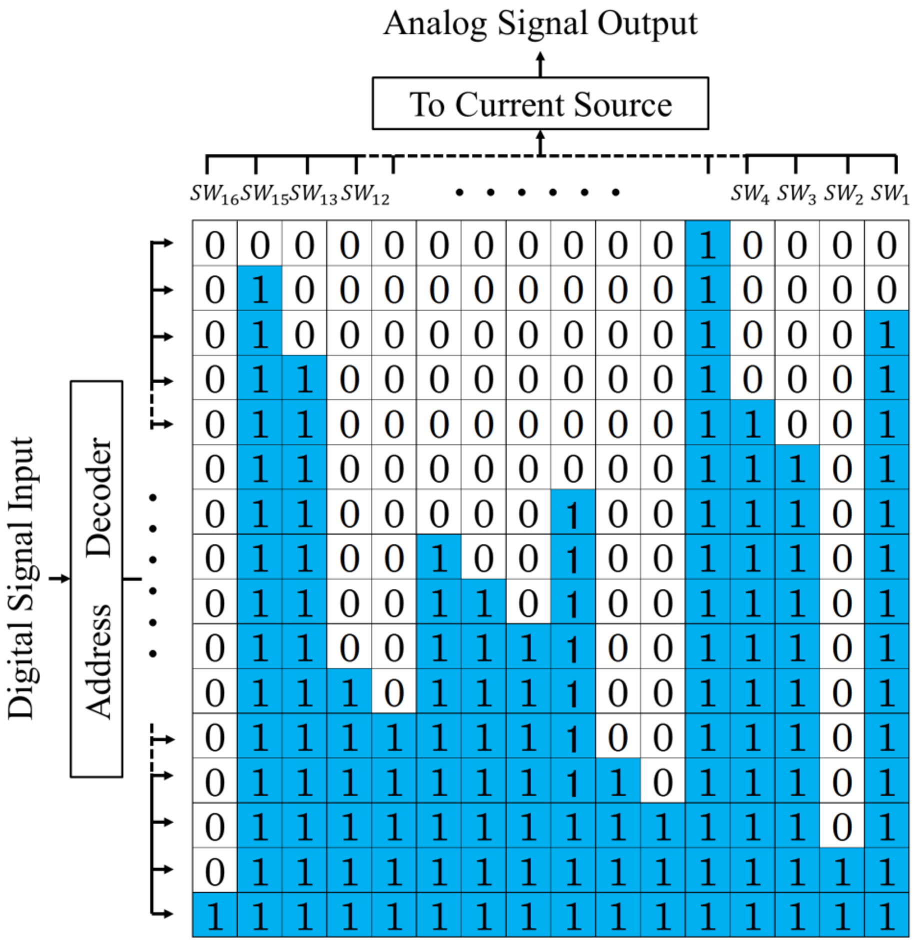
Proposed Algorithm LUT-Magic Square Decoder

- Magic square switching sequence is stored in decoder
- Cancel mismatch effect

1	15	14	4
12	6	7	9
8	10	11	5
13	3	2	16

I_1	I_2	I_3	I_4
I_5	I_6	I_7	I_8
I_9	I_{10}	I_{11}	I_{12}
I_{13}	I_{14}	I_{15}	I_{16}





Analog Signal Output
To Current Source
SW_{16} SW_{15} SW_{13} SW_{12} · · · · · · SW_4 SW_3 SW_2 SW_1
Digital Signal Input
Address Decoder

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

OUTLINE

- Research Objective
- Current Steering DAC
- What is Magic Square ?
- Proposed Algorithm
- Simulation Results
- Conclusion

Simulation Conditions

- MATLAB simulation

- 8-bit unary DAC

  - Static performance (INL, DNL)
  - Dynamic performance (SFDR)

- Compared two methods

  - Conventional thermometer-code decoder usage
  - Proposed magic-square-based algorithm

- Mismatch of current sources

  - Current sources have average of value 1.0
  - Random number between -1 < mismatch < +1 (uniform distribution)

Simulation Conditions

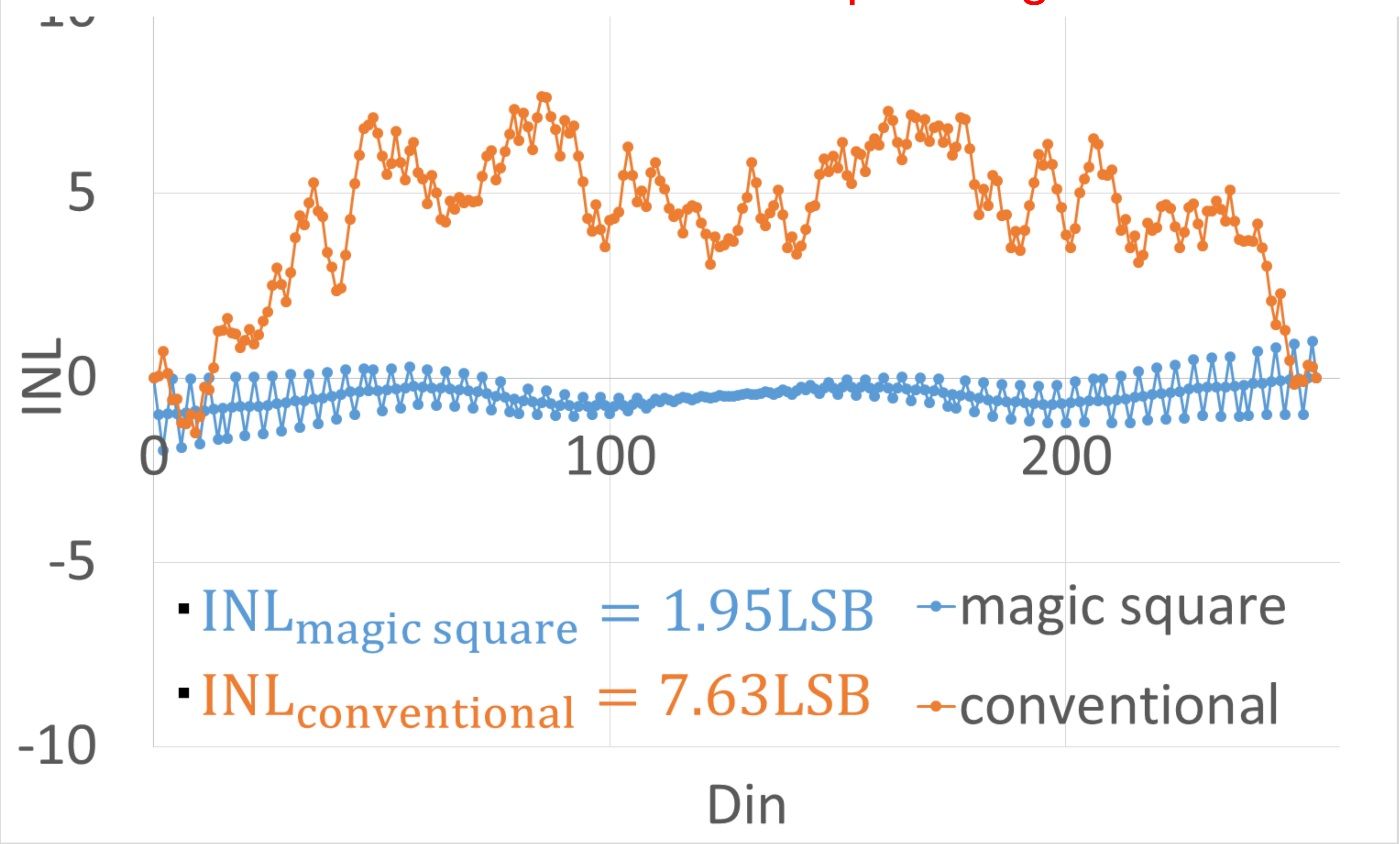
- 使用した16次方陣（8bit）

256	2	3	253	252	6	7	249	248	10	11	245	244	14	15	241
17	239	238	20	21	235	234	24	25	231	230	28	29	227	226	32
33	223	222	36	37	219	218	40	41	215	214	44	45	211	210	48
208	50	51	205	204	54	55	201	200	58	59	197	196	62	63	193
192	66	67	189	188	70	71	185	184	74	75	181	180	78	79	177
81	175	174	84	85	171	170	88	89	167	166	92	93	163	162	96
97	159	158	100	101	155	154	104	105	151	150	108	109	147	146	112
144	114	115	141	140	118	119	137	136	122	123	133	132	126	127	129
128	130	131	125	124	134	135	121	120	138	139	117	116	142	143	113
145	111	110	148	149	107	106	152	153	103	102	156	157	99	98	160
161	95	94	164	165	91	90	168	169	87	86	172	173	83	82	176
80	178	179	77	76	182	183	73	72	186	187	69	68	190	191	65
64	194	195	61	60	198	199	57	56	202	203	53	52	206	207	49
209	47	46	212	213	43	42	216	217	39	38	220	221	35	34	224
225	31	30	228	229	27	26	232	233	23	22	236	237	19	18	240
16	242	243	13	12	246	247	9	8	250	251	5	4	254	255	1

Simulation Result
- Static Performance INL -

Integral Non-Linearity (INL)

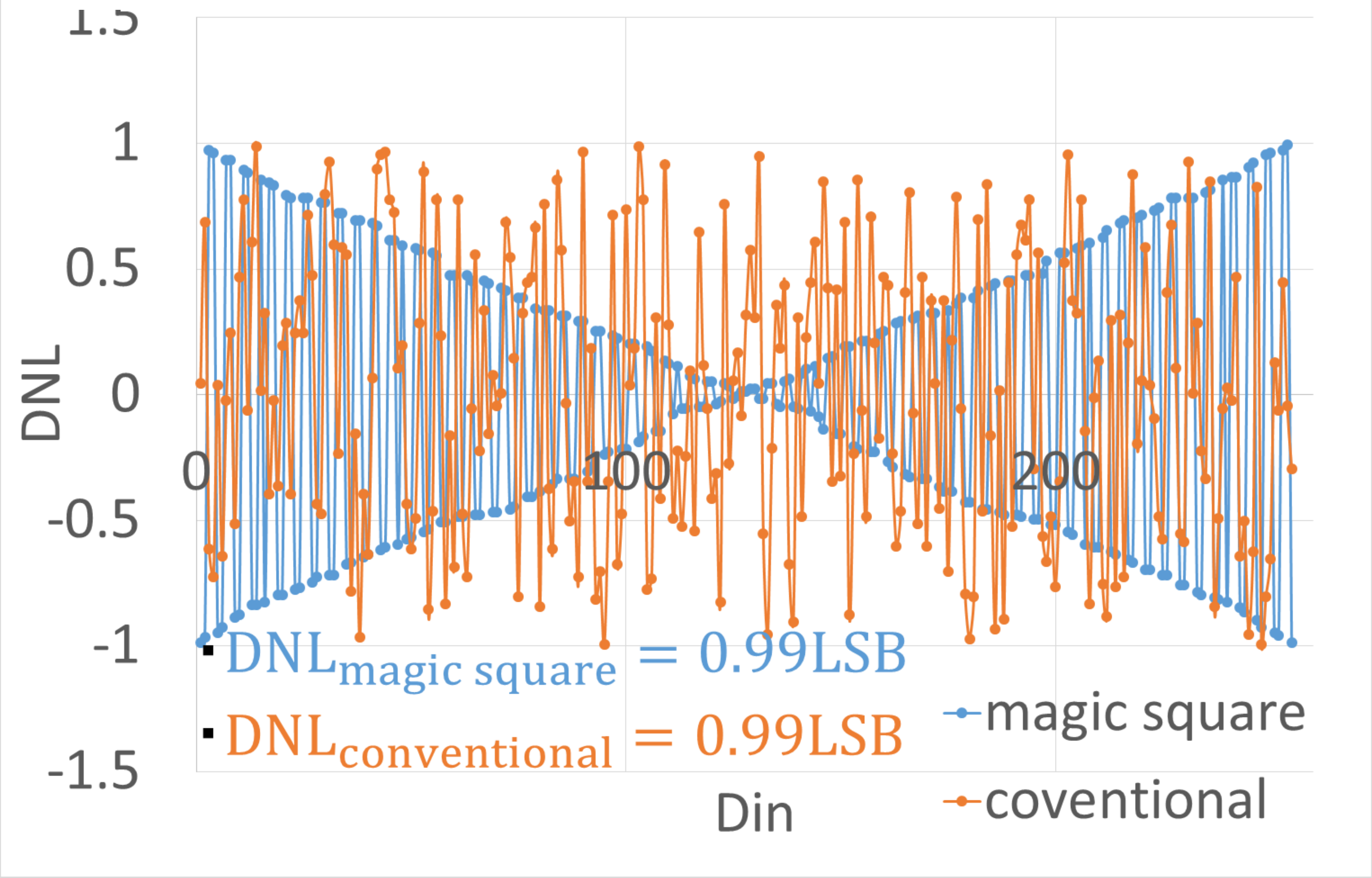
- 5.7 LSB improvement by the magic square algorithm
- 0.0 LSB at the center of the input range Din



Simulation Result
- Static Performance DNL

Differential Non-Linearity (DNL)

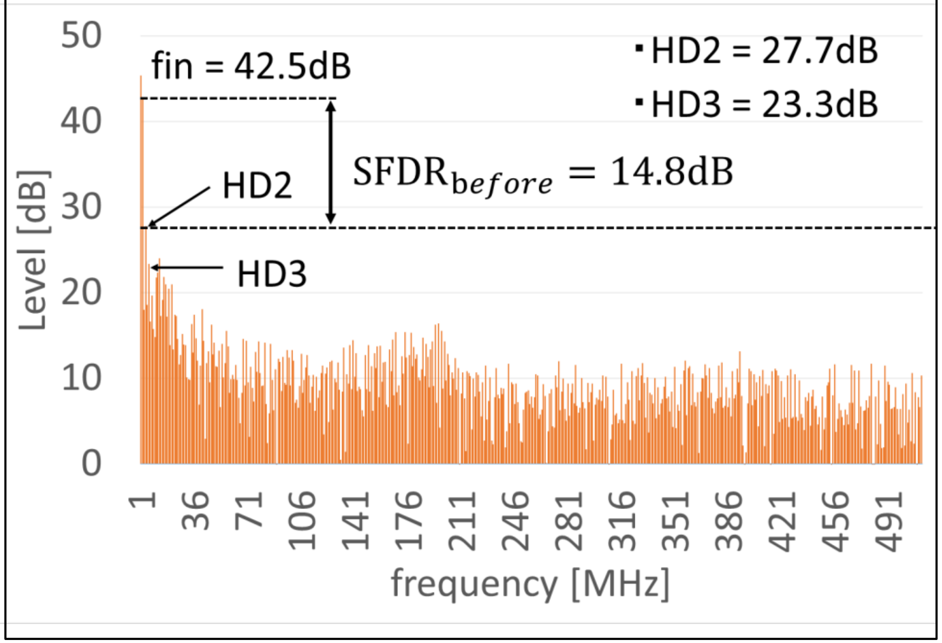
● Small at the center of the input range Din



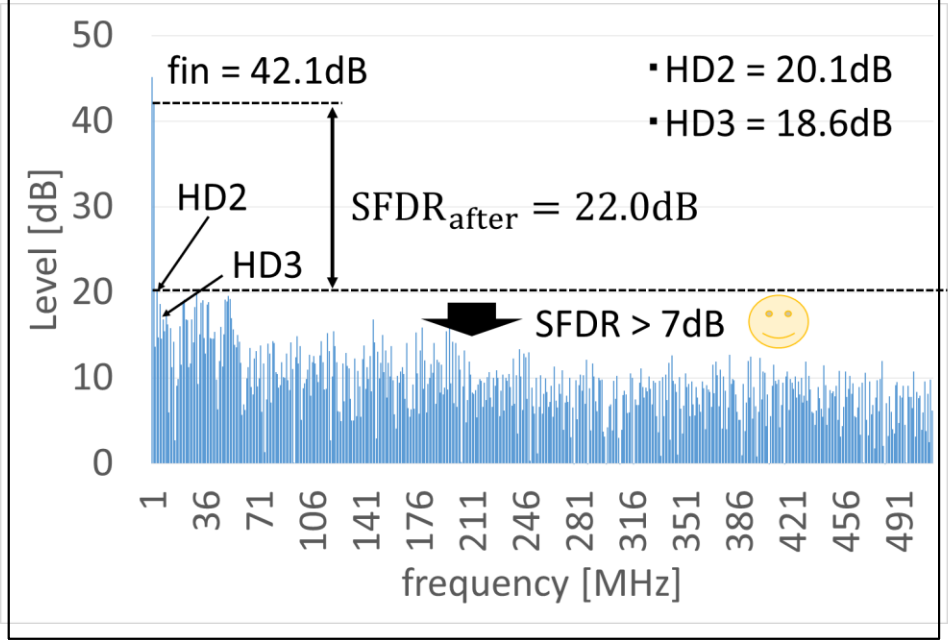
- $\mathrm{DNL}_{\text{magic square}} = 0.99\mathrm{LSB}$
- $\mathrm{DNL}_{\text{conventional}} = 0.99\mathrm{LSB}$

Simulation Result
- Dynamic Performance SFDR -

● Conventional method



● Proposed method



● SFDR improvement by 7 dB

OUTLINE

- Research Objective
- Current Steering DAC
- What is Magic Square ?
- Proposed Algorithm
- Simulation Results
- Conclusion

Conclusion

- Unary DAC linearity improvement

  - Cancel unit current cell mismatch effects

  - Unit current cell selection algorithm

  ➡ Digital method

  - Based on magic square

  - Measurement of the order of current cell values

- MATLAB simulation

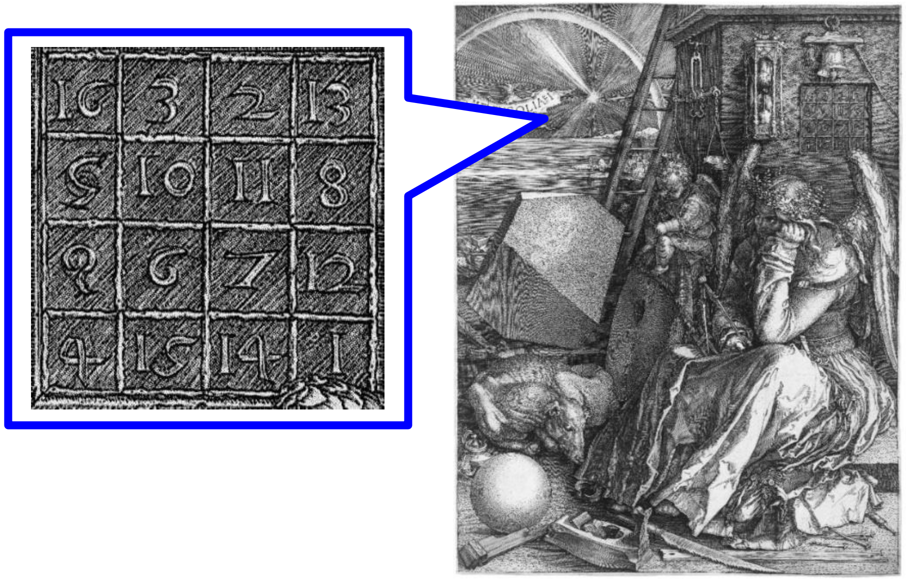
  - INL , DNL improvement
  at the center of the input range.

  - SFDR improvement

Final Statement

温故知新

Classical mathematics can contribute modern technology.



質問内容（1）

- 基本研究の話。先行研究があるか
  →ありません
- 魔方陣がどう有効に働いているか
  →定和性により、ミスマッチが均等に分配され、キャンセルできる
- ２次元に作る必要性は
- 縦方向の定和性が使われていない。
  →入力が多くなれば、縦方向の定和性が効いてくると考えている
- 行単位でバランスをとっている，横方向の並び順は意味があるのか。
  →大小の組み合わせ（魔方陣の配列）になっているので、並び順は意味があると考えている
- 1, 15, 14, 4 の順は関係ある，4つ単位で定和性が活きてきてるんじゃないの
- システマティックなばらつきは考慮しているのか。
  →今回は、考慮していないが、ばらつきに応じてセルを並び替えているので、結果は変わらないと考えられる
- いい並び方が他にある？単純に大小を交互にスイッチングしたほうが良いのでは
- 縦の部分も活きていると感じた。マクロに見れば活きていないけど。
- I1, I2, I3の意味，スライドの書き方を丁寧に。混乱してしまう。
- 異なる魔方陣でやってみたか
- 別の乱数でやってみたか

質問内容（2）

- いい並び方が他にある？単純に大小を交互にスイッチングしたほうが良いのでは
  →魔方陣が完璧な方法ではないので、検討課題ではある。
- 縦の部分も活きていると感じた。マクロに見れば活きていないけど。
- I1, I2, I3の意味，スライドの書き方を丁寧に。混乱してしまう。
- 異なる魔方陣でやってみたか
  →4通りやった
- 別の乱数でやってみたか
  →yes