Study of Gray Code Input DAC for Glitch Reduction

群馬大学
GUNMA UNIVERSITY

***Adhikari Gopal Richen Jiang Haruo Kobayashi**

Kobayashi Laboratory, Gunma University, Japan

Outline

- Research Objective
- Introduction to DAC
- Glitches
- Introduction to Gray Code
- Gray Code Input DAC
  - Switch Matrix Design
  - Voltage Mode Gray Code Input DAC
  - Current Steering Mode Gray Code Input DAC
- Conclusion

Research Objective

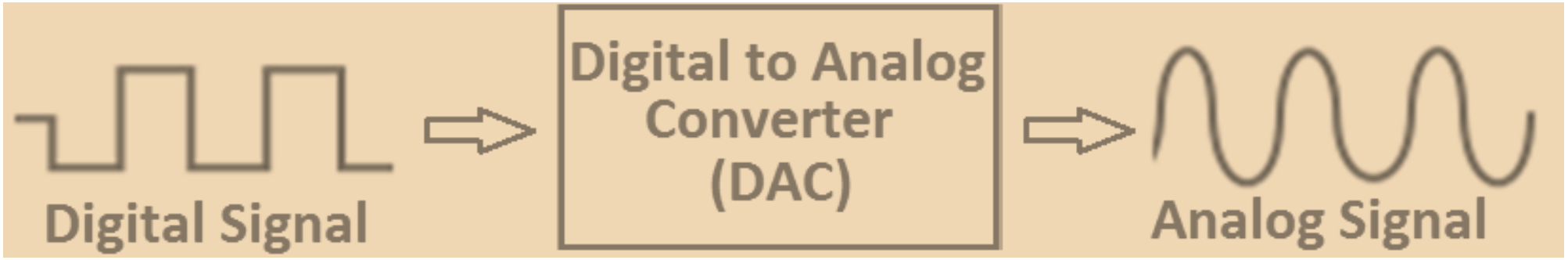
- ✓ Research and implement DAC for glitch reduction using Gray code input
  - *(difficult to design)

Approach

- ✓ Use MOSFETs for DAC design
- ✓ Utilization of Gray code input for glitch reduction

Introduction to DAC

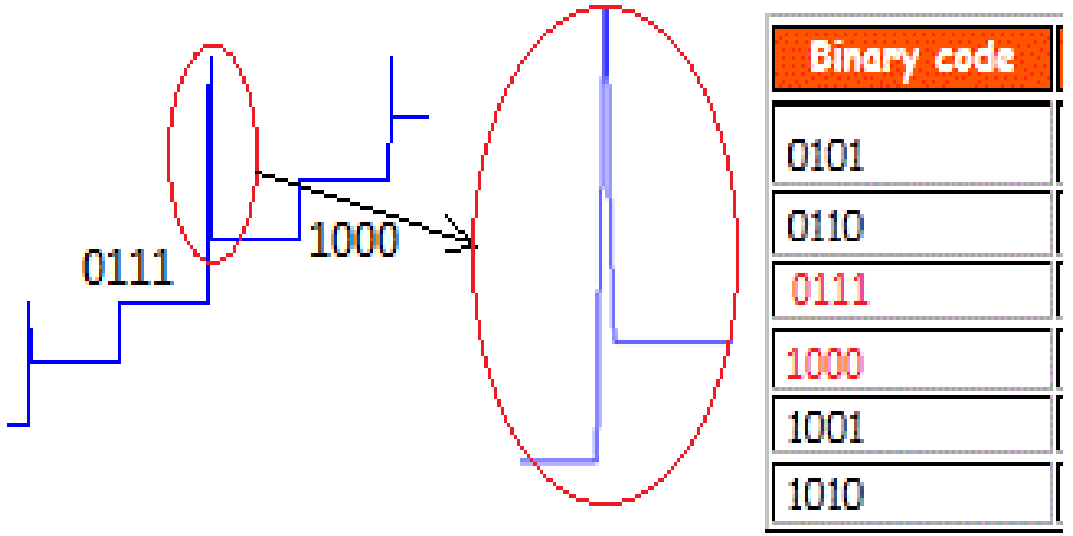
- Convert digital signal to analog signal



- Signal to be recognized by human senses
- Widely used in signal processing

What are Glitches ?

- ✓ Voltage spikes
- ✓ Reasons for glitches
  - Capacitive coupling
  - Differences in switching



- ➢ Glitch behaviour → Dominated by difference in switching
- ➢ Switching of MSB → Most significant glitches

(Multiple switches changing states at once)

Glitch Problem and Remedy

Effects of Glitch

- Serious deterioration of images, videos and sounds

Remedy

- Using high-order reconstruction filter
- Using track/hold circuitry at the DAC output

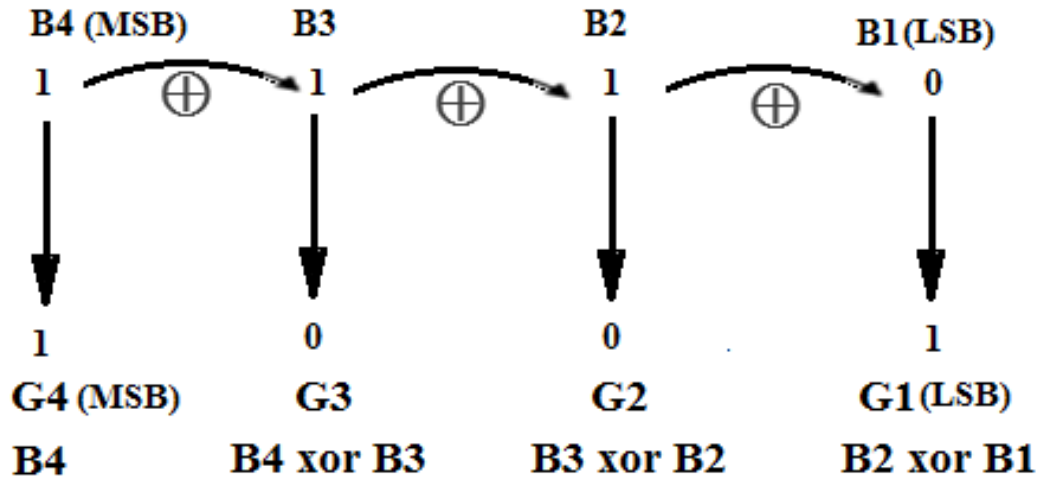
} Extra Space in IC, Expensive

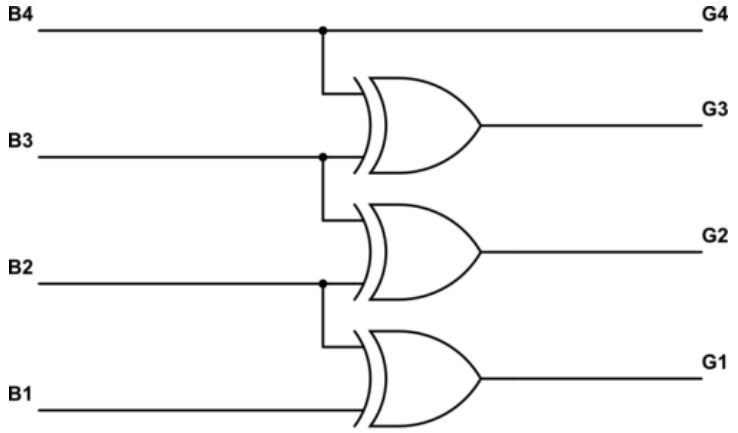
- **Using Gray code input DAC topologies** ✓

What is Gray code ?

- Gray code → Alternative representation of binary code
- Two adjacent numbers→ Only one bit change
- Reflected binary code

Binary to Gray code conversion





Binary to Gray code converter

Gray Code versus Binary Code

Binary code Multiple bits change at a time

Trigger more switches

Example. 1 → 2 --- 0001 → 0010 2 bits change

7 → 8--- 0111 →1000 all 4 bits change

Gray code Only one bit changes at a time

Triggers one switch

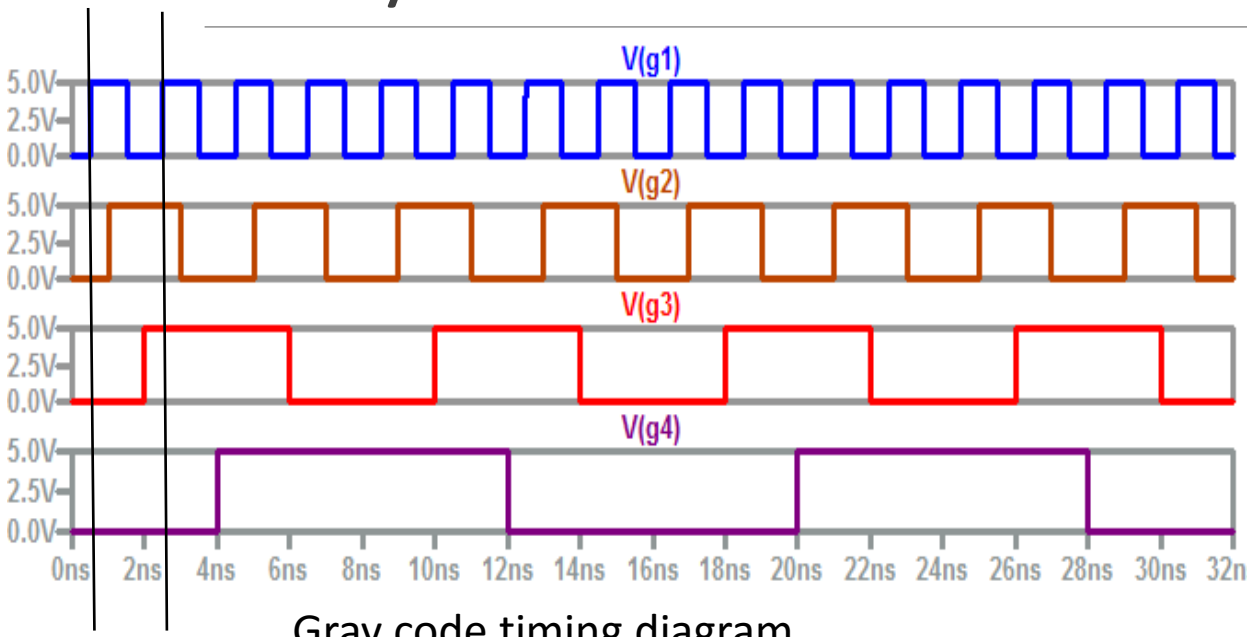
Example. 1 → 2 --- 0001 → 0011 one bit change (B2)

7 → 8 --- 0100→1100 one bit change (B4)

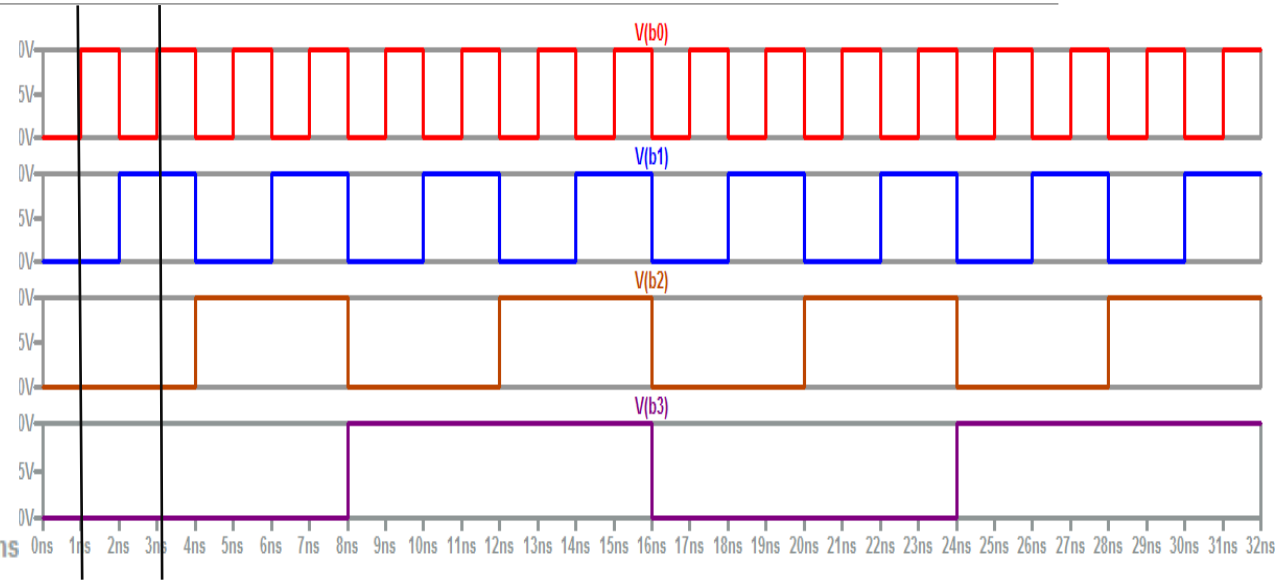
Less glitches

Decimal	Binary	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Gray code versus Binary code Timing Diagram



Gray code timing diagram



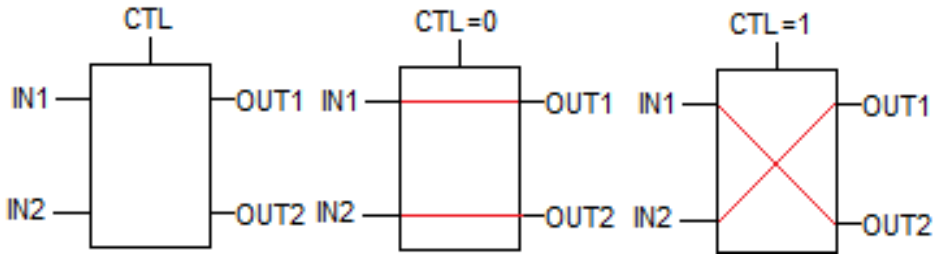
Binary code timing diagram

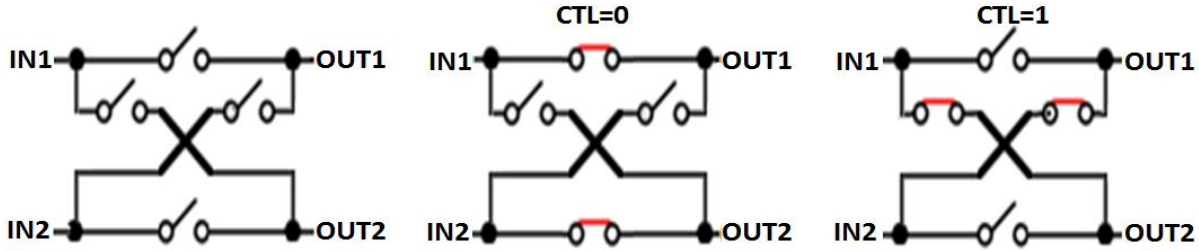
Gray code → Only one bit changes at a time 0001→0011
Binary code → Multiple bits change at a time 0001→0010

Using Gray code → Less glitches expected to appear

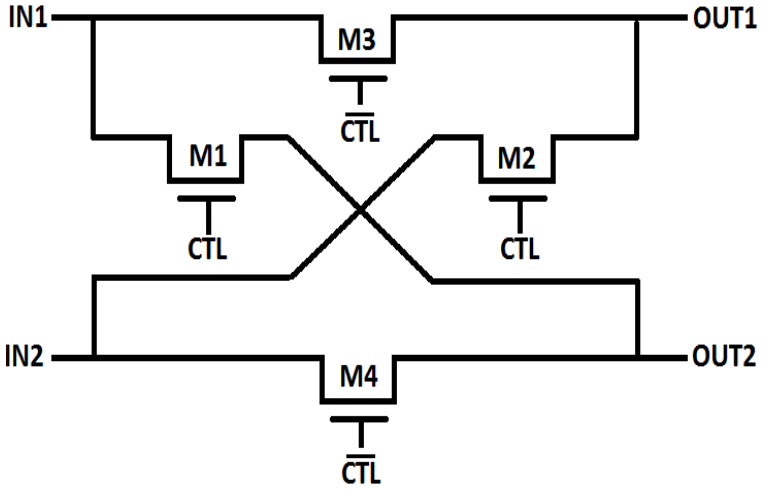
Gray Code Input DAC

Switch Matrix Design

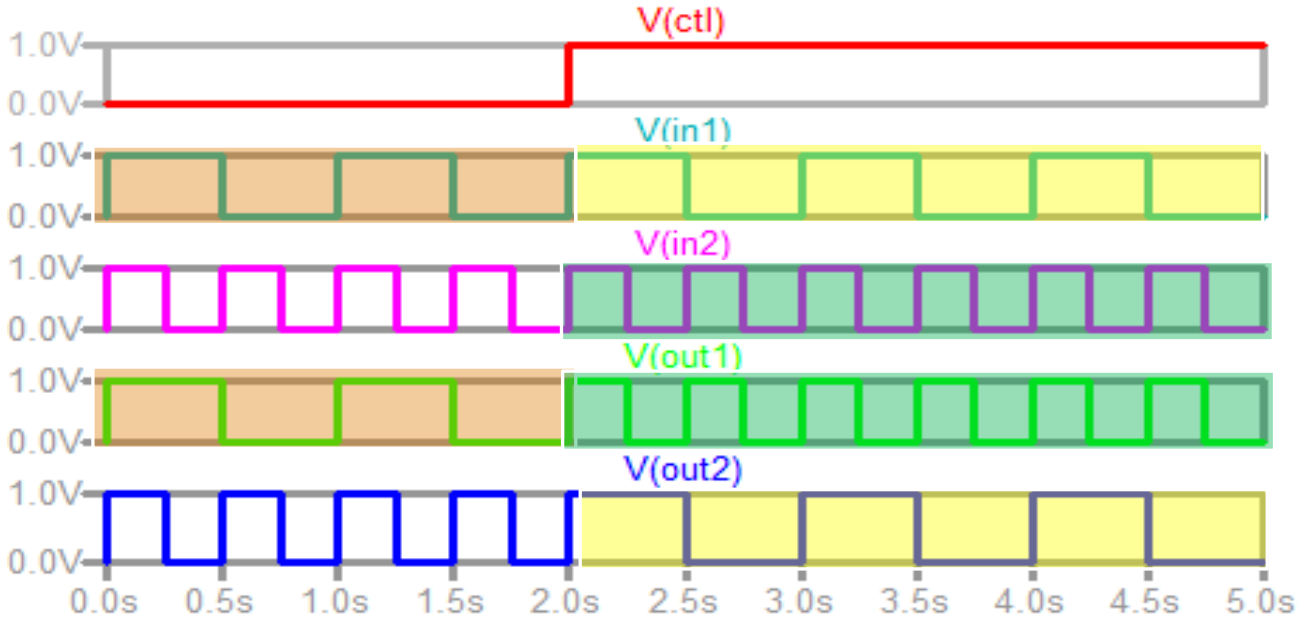


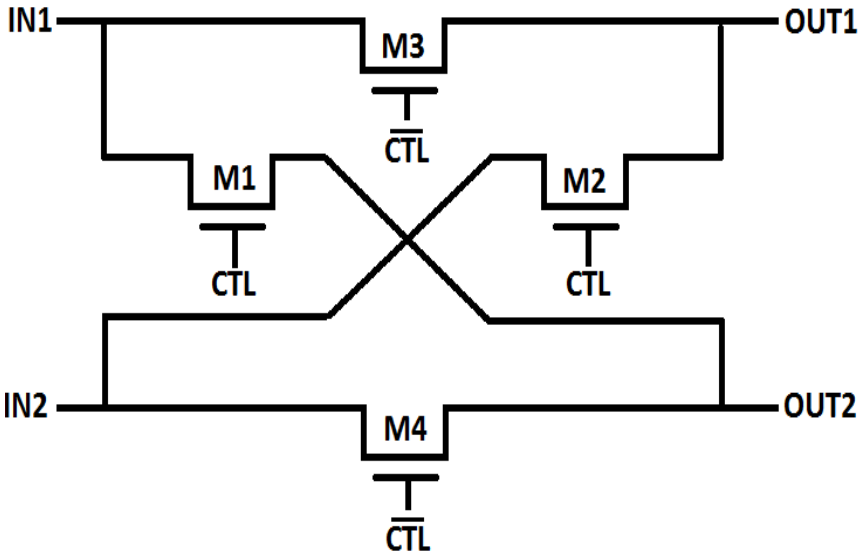


Switch is DPDT (Double-Pole Double-Throw)



Switch Matrix Operation





CTL → LOW:

M3 , M4 → ON, M1, M2 → OFF

IN1 = OUT1, IN2 = OUT2

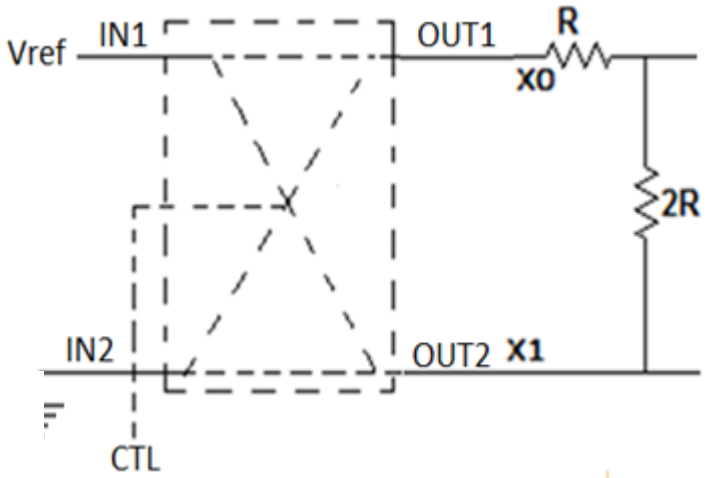
CTL → HIGH:

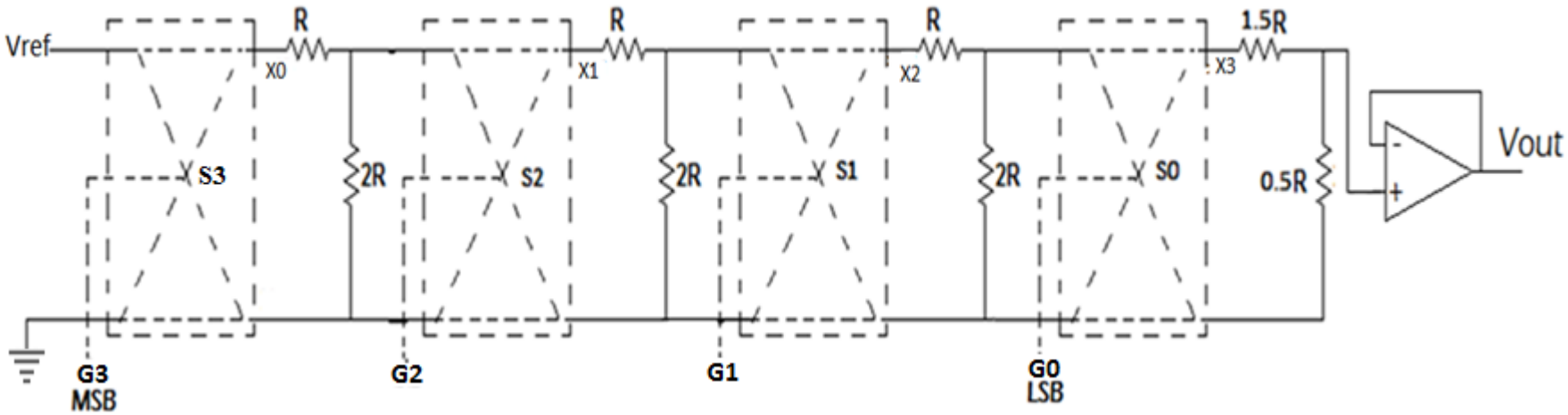
M1, M2 → ON, M3, M4 → OFF

IN1 = Out2 , IN2 =OUT1

Voltage Mode Gray Code Input DAC

- IN1 = Vref
- IN2 = 0
- CTL ← Gray code input
- OUT1, OUT2 → Connected with R-2R Ladder





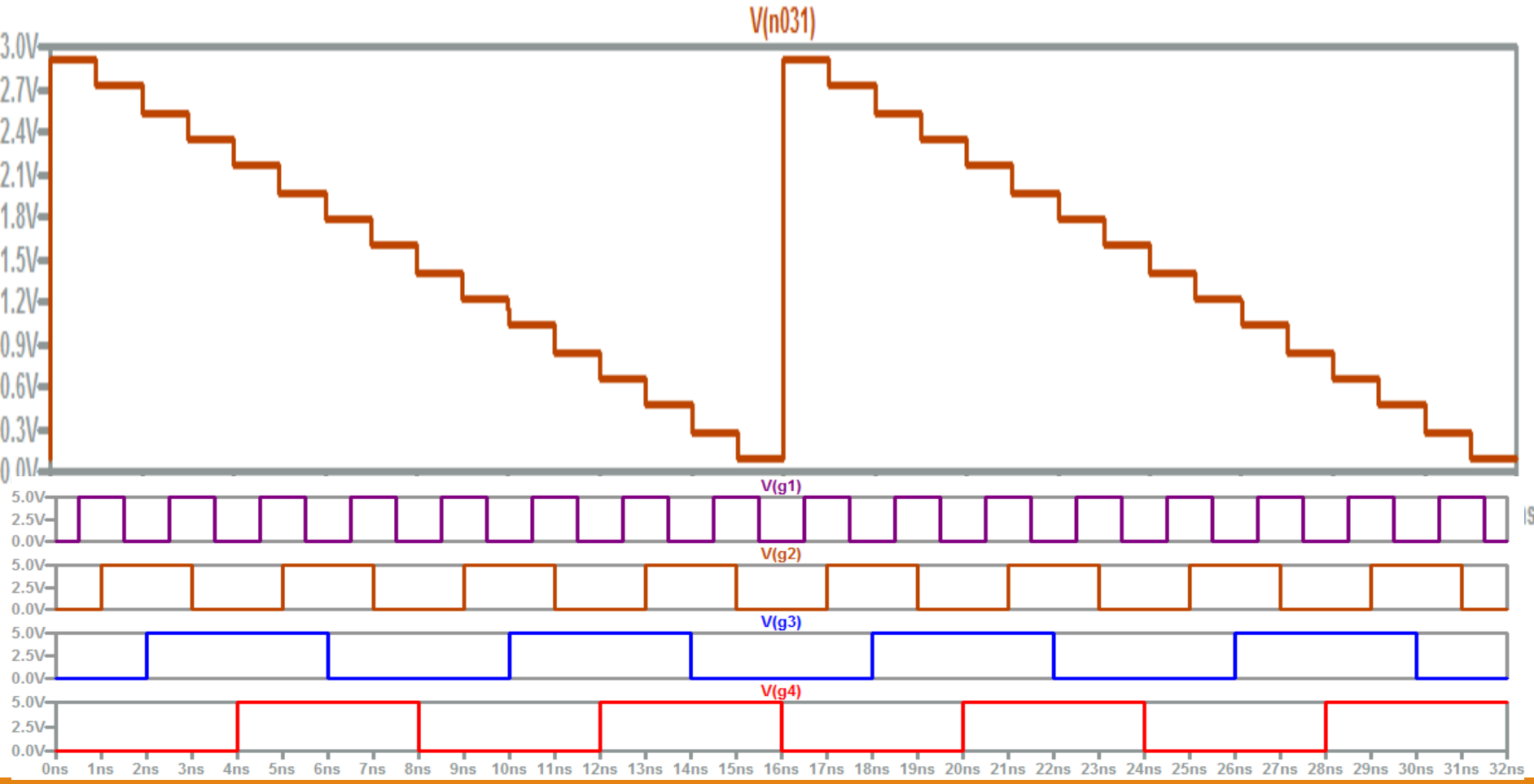
- Final stage → terminated with 1.5R, 0.5R resistors.

$$Vout(D) = \frac{Vref}{2^{n+1}} |(2D - 1)|$$

n : number of bits
D =1, 2, 3...n+1

Voltage Mode Gray Code Input DAC SPICE Simulation Results

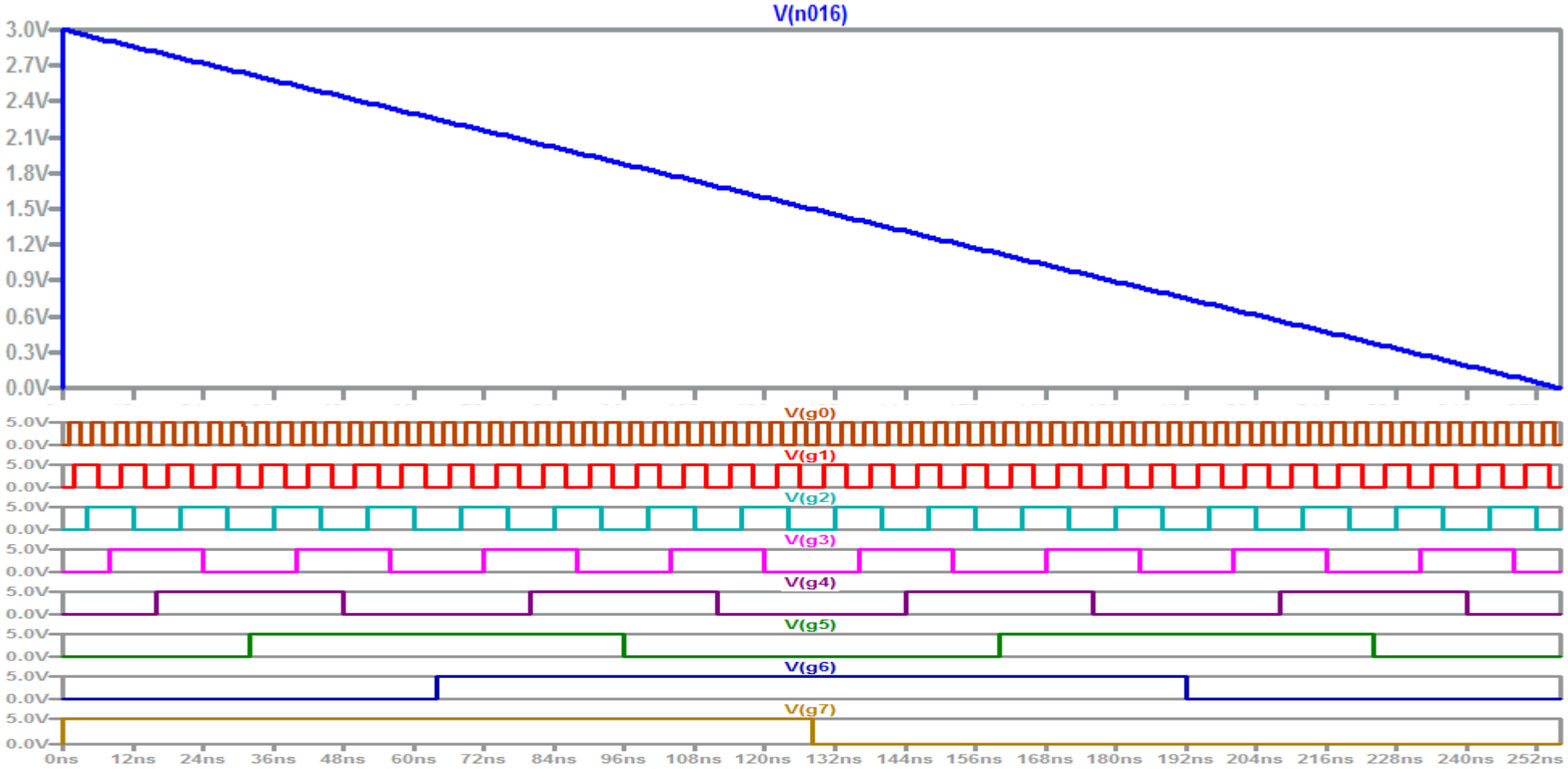
4-bit case



D	Bits	Vout	
1	0000	3/32	0.09375
2	0001	9/32	0.28125
3	0011	15/32	0.46875
4	0010	21/32	0.65625
5	0110	27/32	0.84375
6	0111	33/32	1.03125
7	0101	39/52	1.21875
8	0100	45/32	1.40625
9	1100	51/32	1.59375
10	1101	57/32	1.78125
11	1111	63/32	1.96875
12	1110	69/32	2.15625
13	1010	75/32	2.34375
14	1011	81/32	2.53125
15	1001	87/32	2.71875
16	1000	93/32	2.90625

Voltage Mode Gray Code Input DAC SPICE Simulation Results

8-bit case

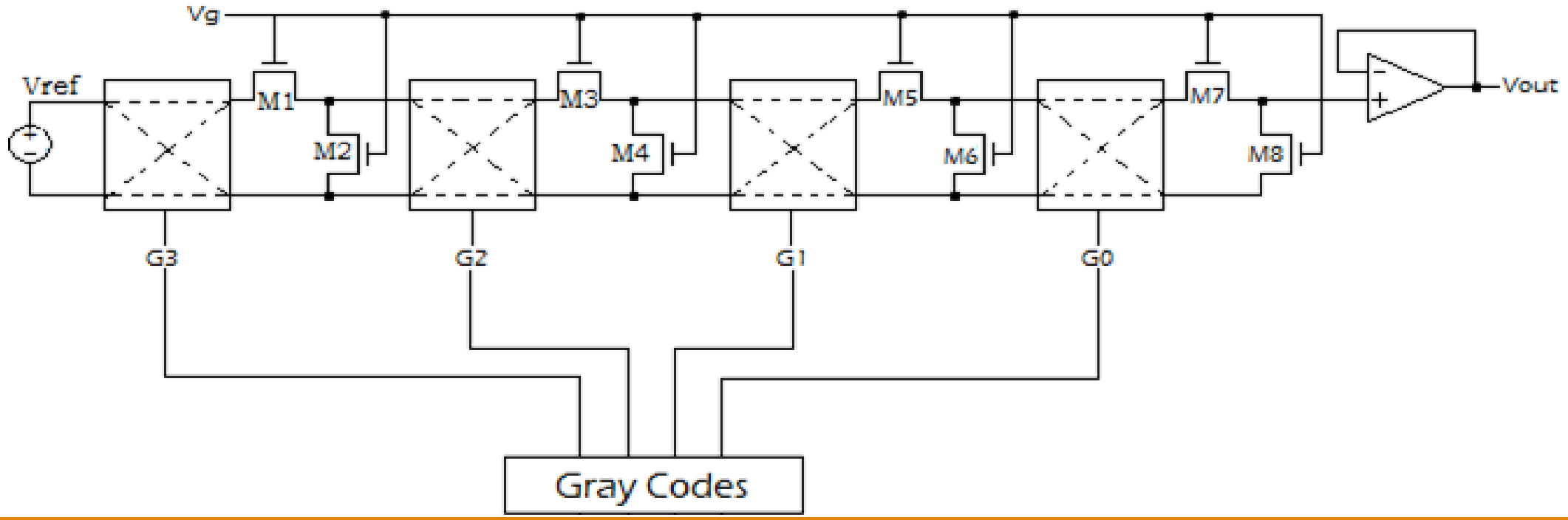


NO GLITCHES

Voltage Mode Gray Code Input DAC MOSFET Implementation

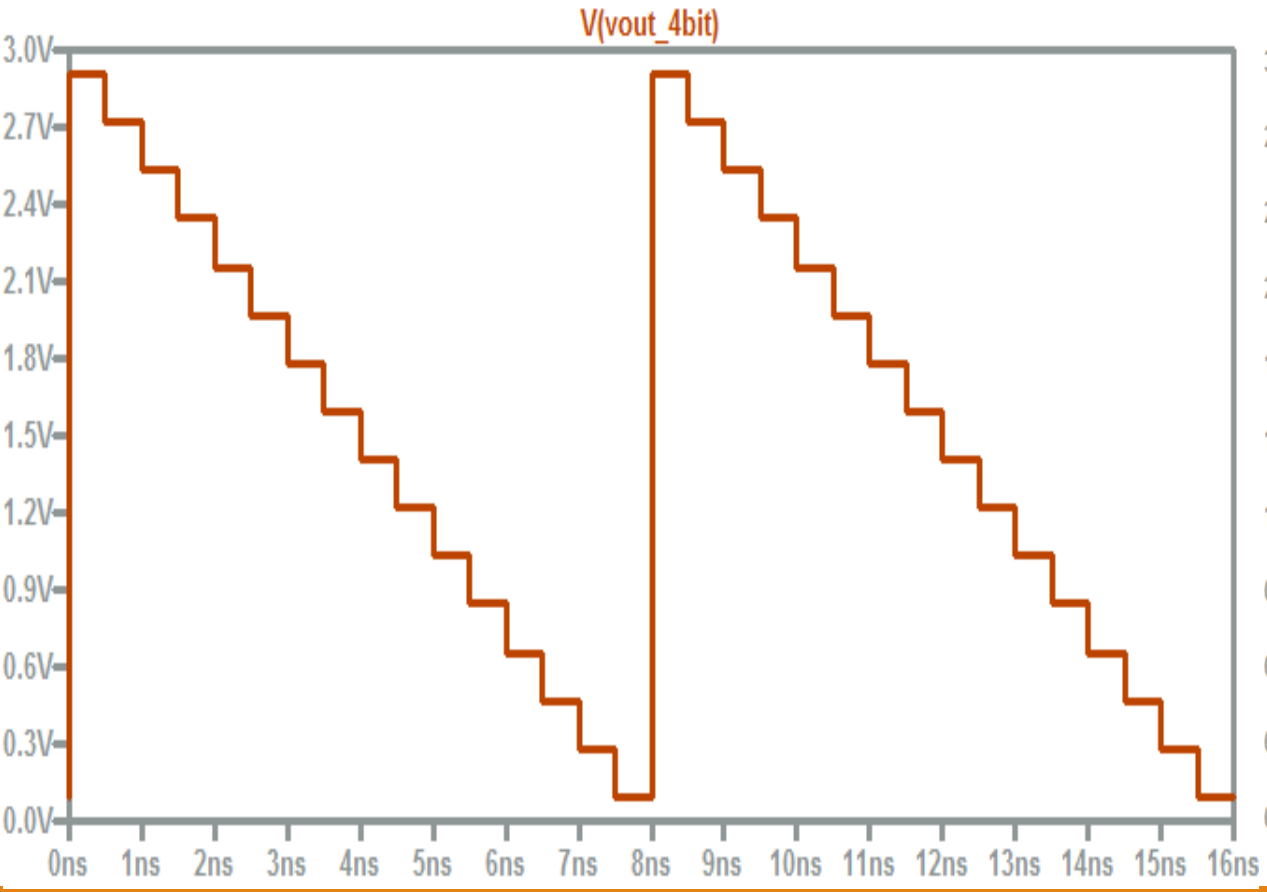
Aspect ratios W/L for R, 2R, 1.5R, 0.5R

$$R = \frac{V_{DS}}{Id_{sat}} = \frac{V_{DS}}{\frac{u_n C_{ox}}{2} \times \left(\frac{W}{L}\right) \times (V_{GS} - V_{TH})^2}$$

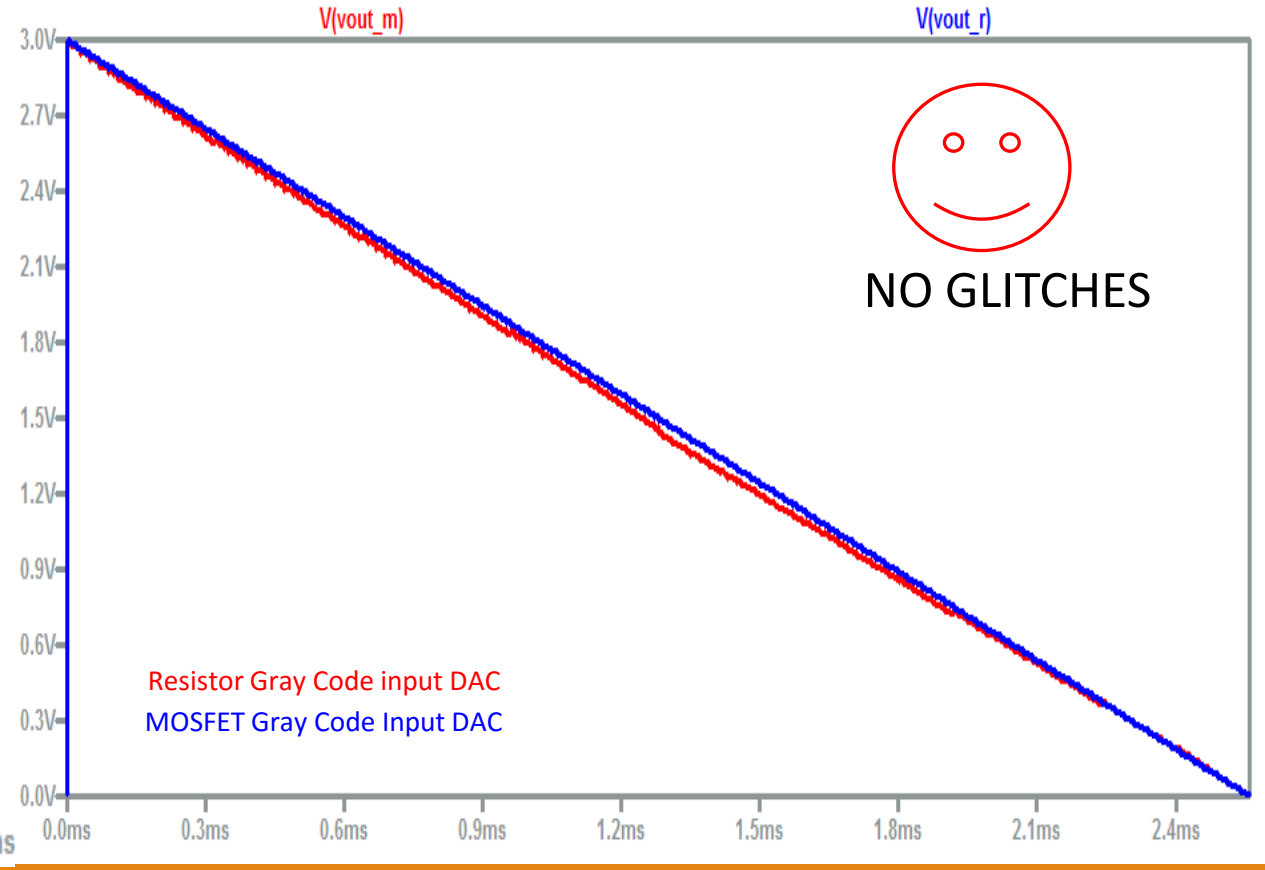


Voltage Mode Gray Code Input DAC MOSFET Implementation Simulation Results

4-bit case

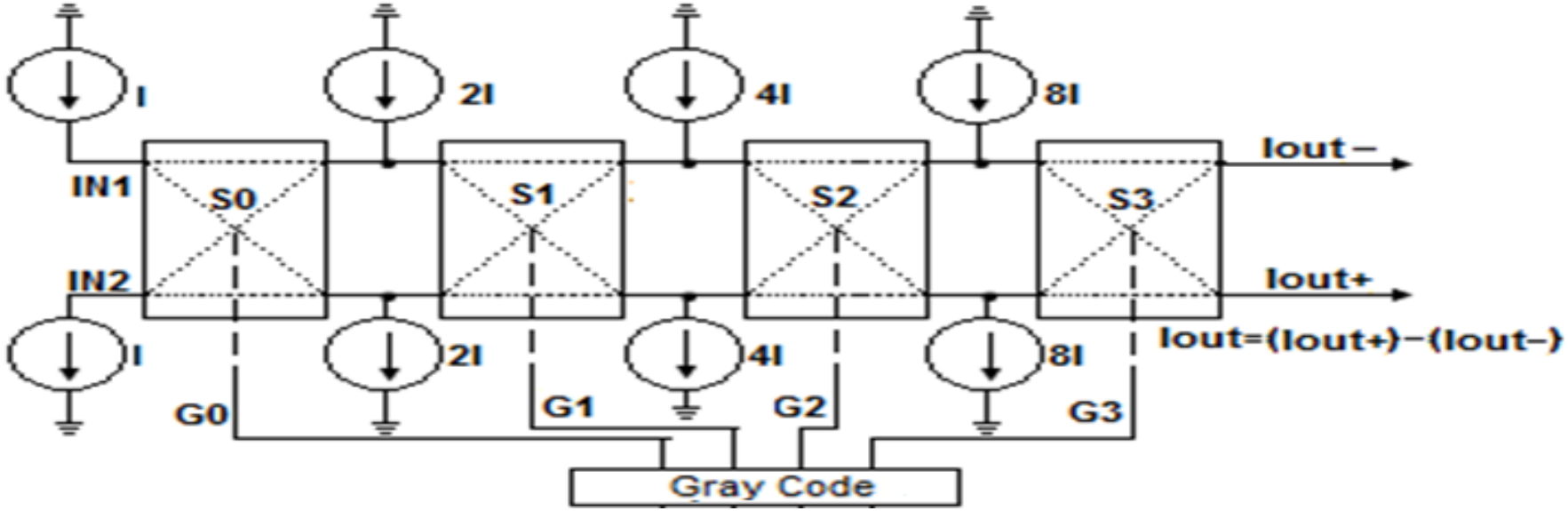


8-bit case



Current Steering Mode Gray Code Input DAC
Circuit Configuration

- IN1, IN2, intermediate stages →binary weighted current sources.
- Gray code alters the way the switches are triggered
- Iout=Iout+ – Iout –



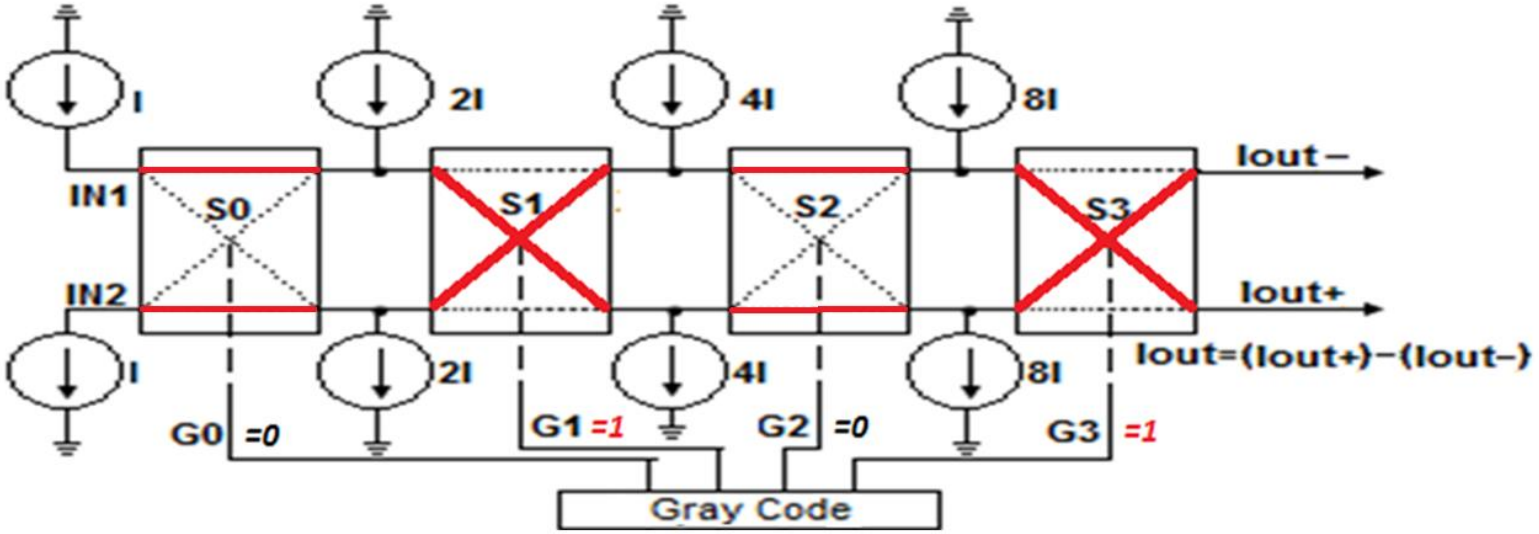
Current Steering Mode Gray Code Input DAC
Circuit Operation

For 1010 Gray code,

S3, S1 → ON, the other switches → OFF

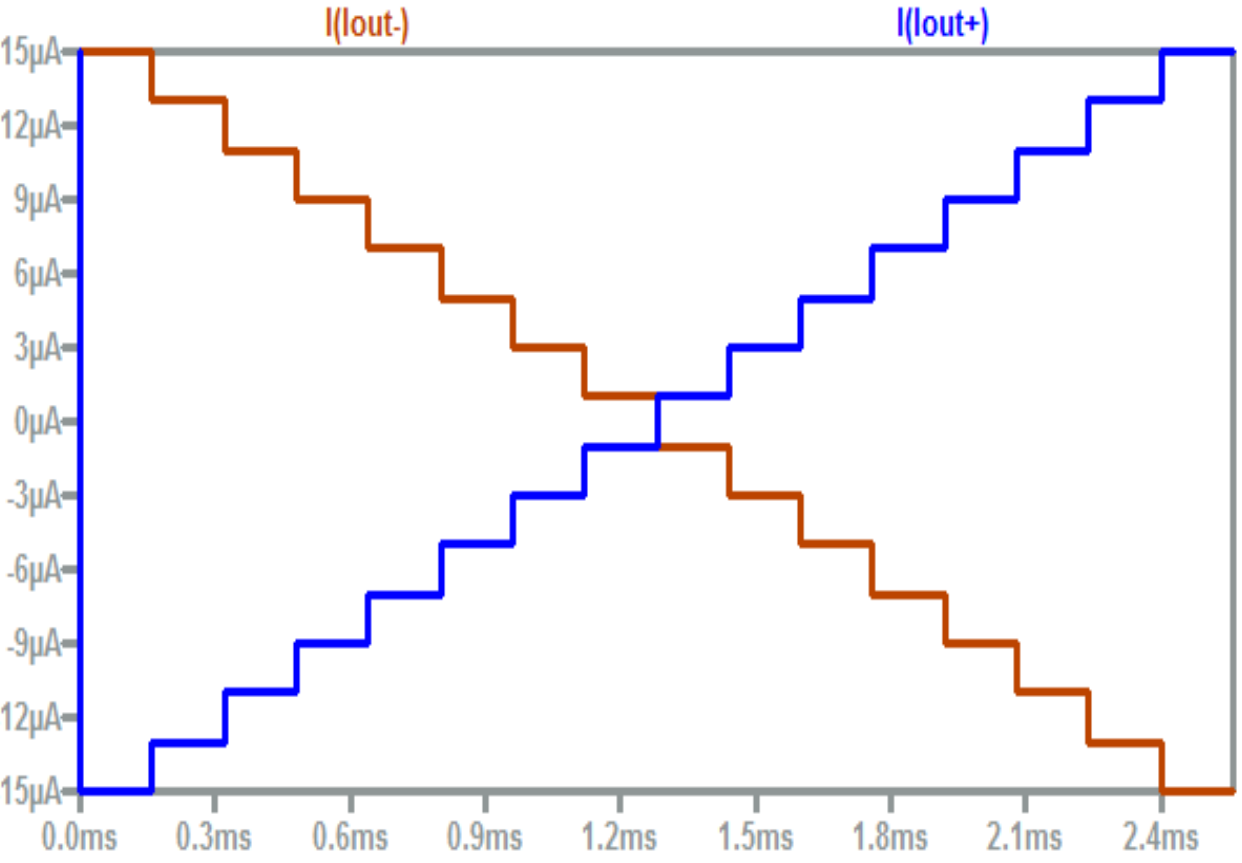
$$Iout -= I + 2I - 4I - 8I = -9I$$

$$Iout += -I - 2I + 4I + 8I = 9I$$

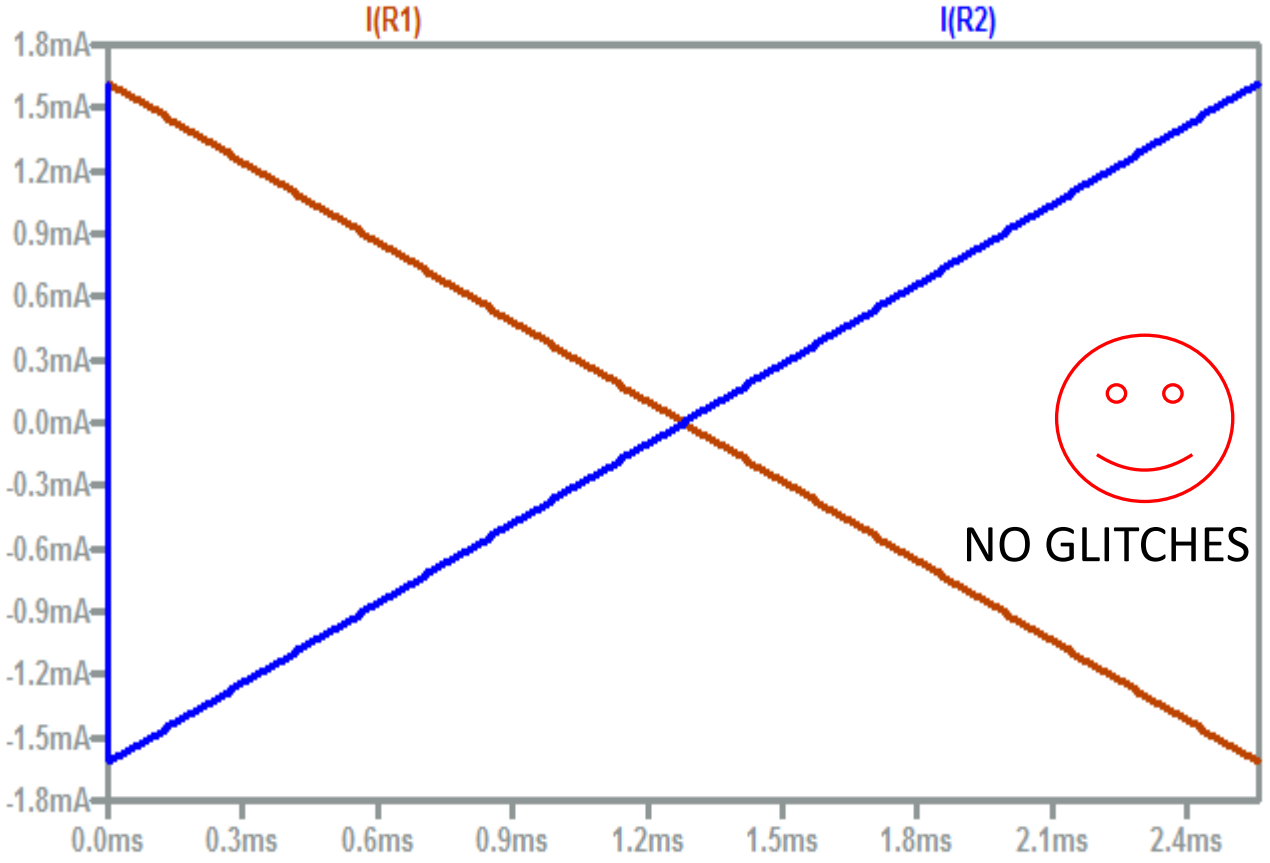


Current Steering Mode Gray Code Input DAC SPICE Simulation Results

4-bit case



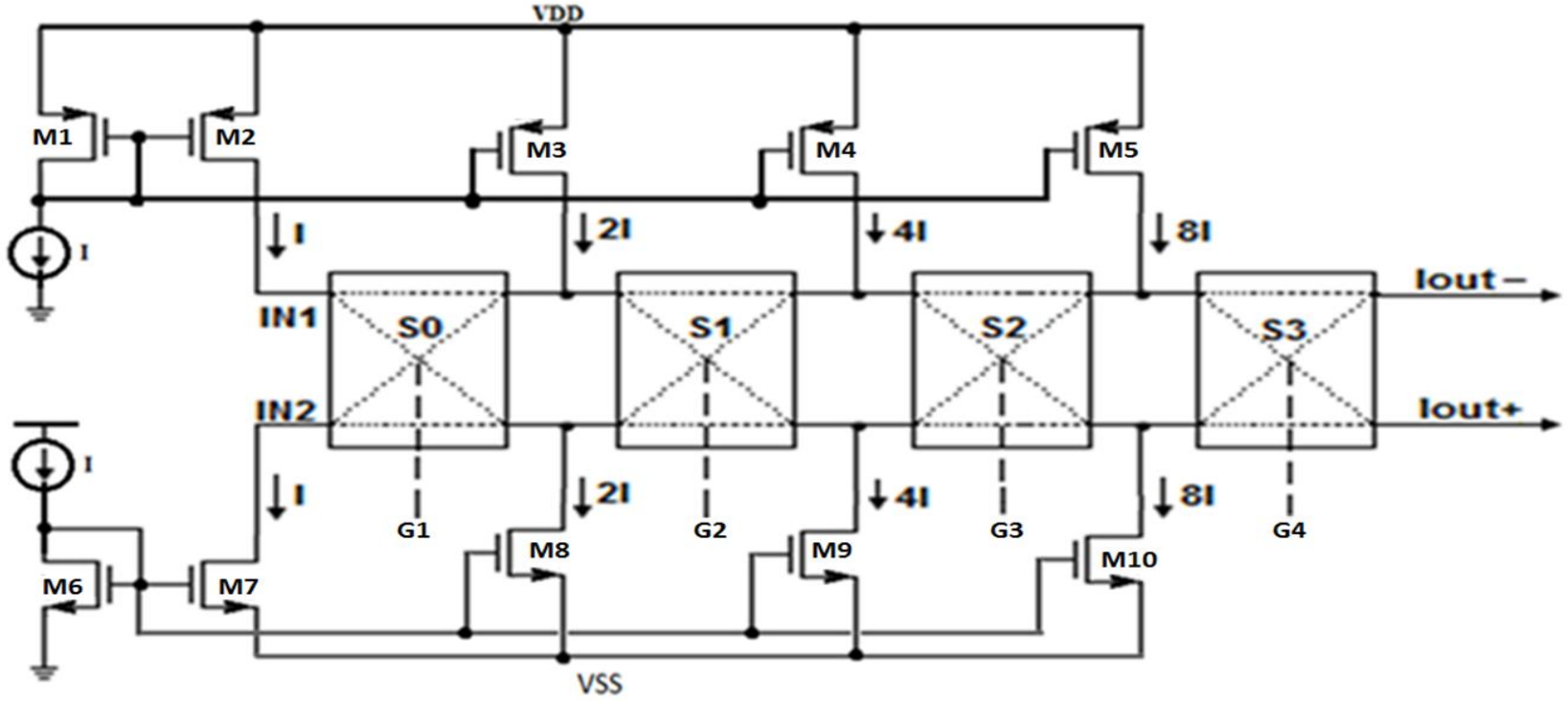
8-bit case



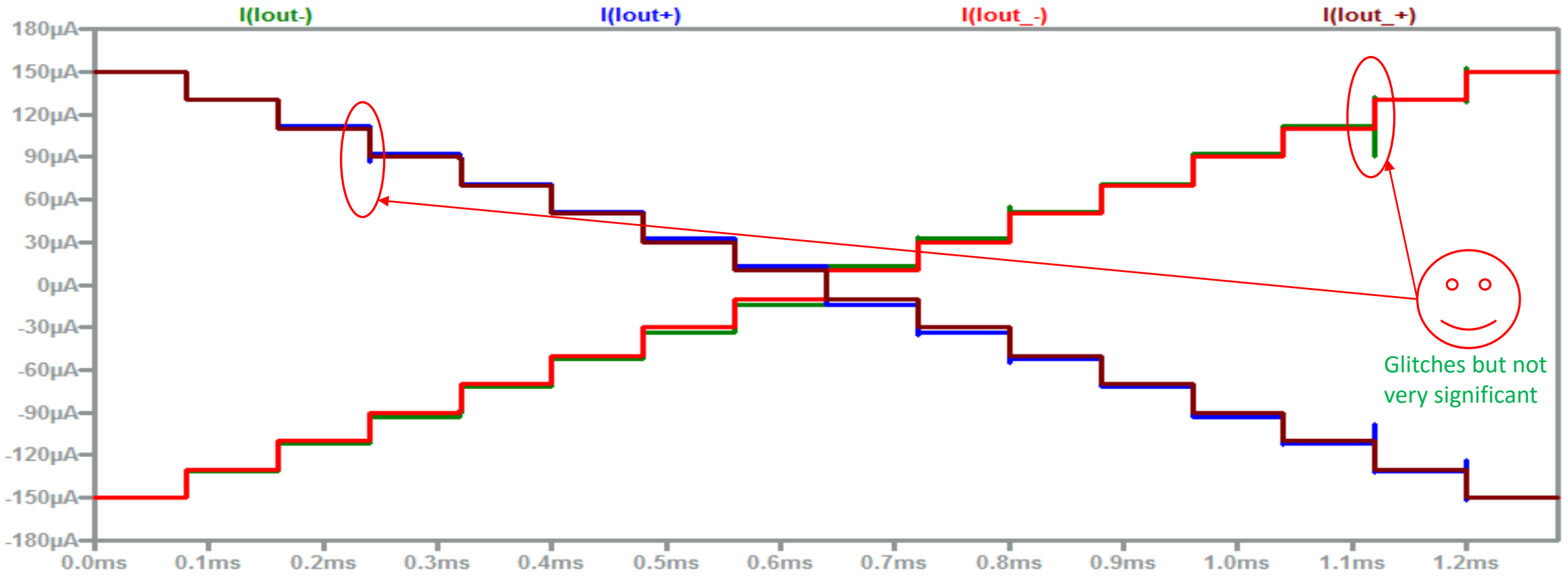
Current Steering Mode Gray Code Input DAC MOSFET Implementation

$M2,\ M3,\ M4,\ M5$ generate $I,\ 2I,\ 4I, 8I$ (current source)

$M7,\ M8,\ M9, M10$ generate $I,\ 2I,\ 4I, 8I$ (current sink)

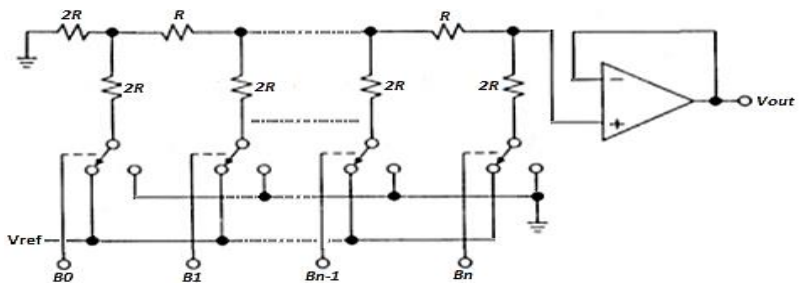


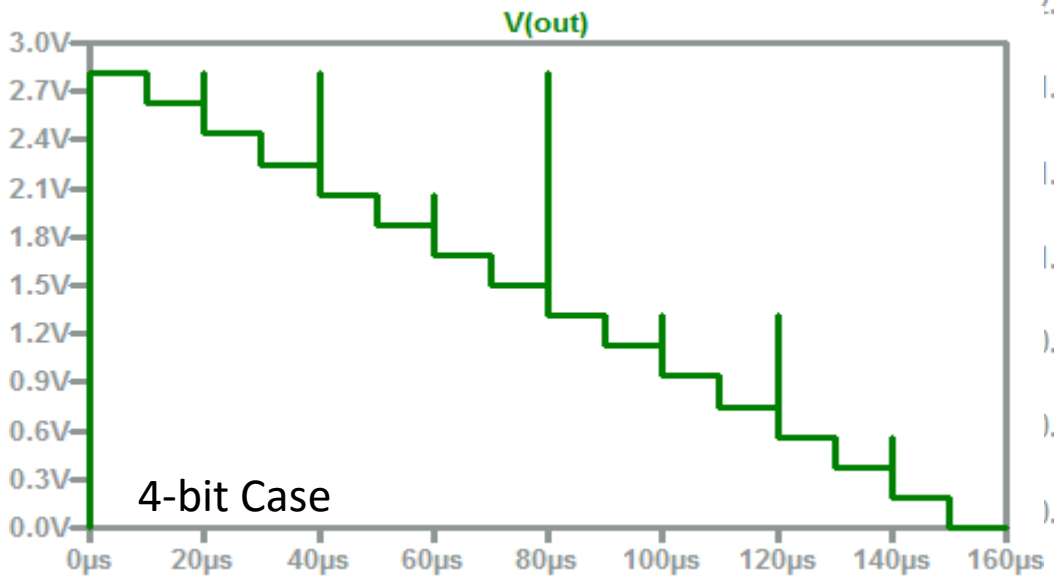
Current Steering Mode Gray Code Input DAC MOSFET Implementation Simulation Results



These glitches are due to mismatch of PMOS and NMOS

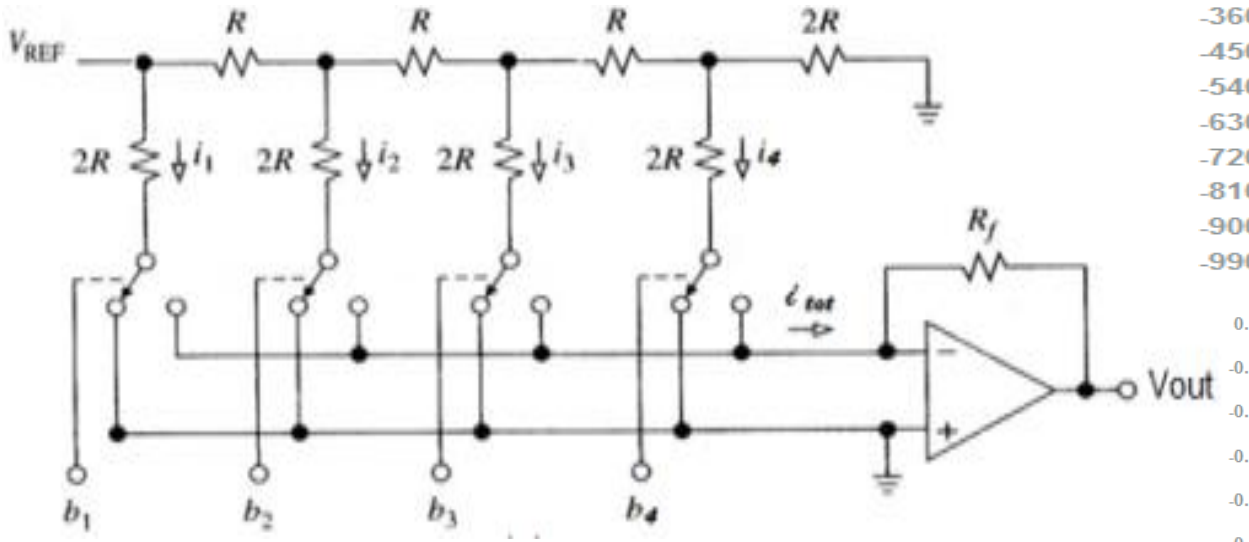
R-2R Binary Voltage Mode DAC

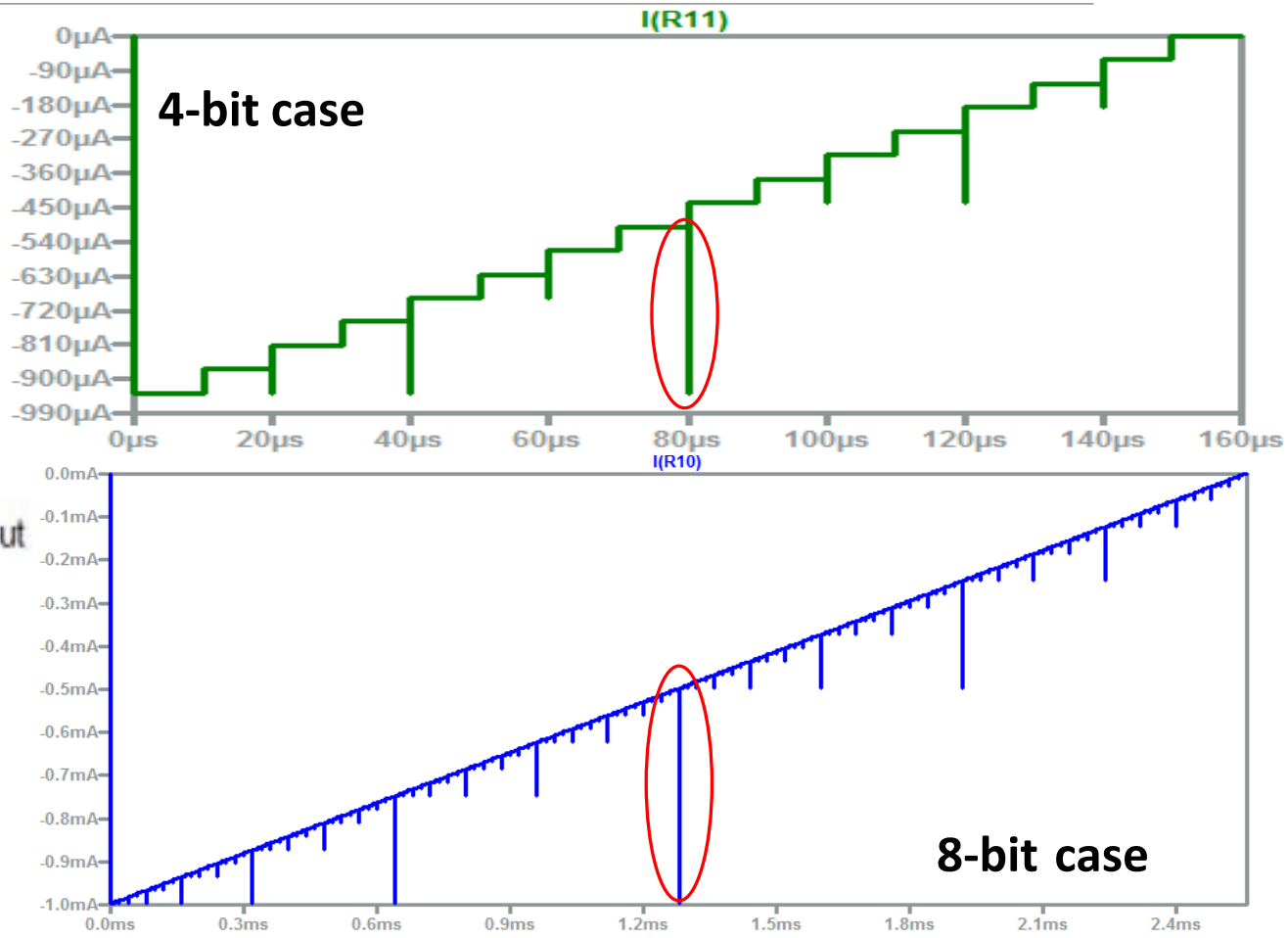




Glitches get introduced in binary code R-2R DAC

R-2R Binary Current Mode DAC





Conclusion

- ✓ R-2R DACs prone to glitches → Multiple bits switching at a time.
- ✓ Claims of Gray code DACs being difficult to design

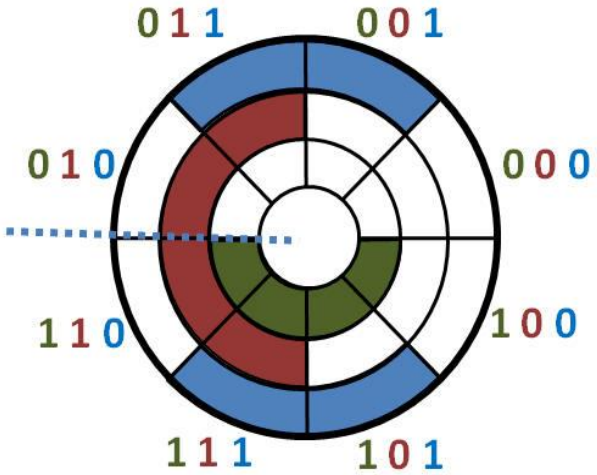
  but successfully designed and simulated
  - ✓ Gray code Input DACs reduce glitches considerably
  - ✓ No extra space needed for IC
  - ✓ No extra circuit needed to remove glitches

Final Statement

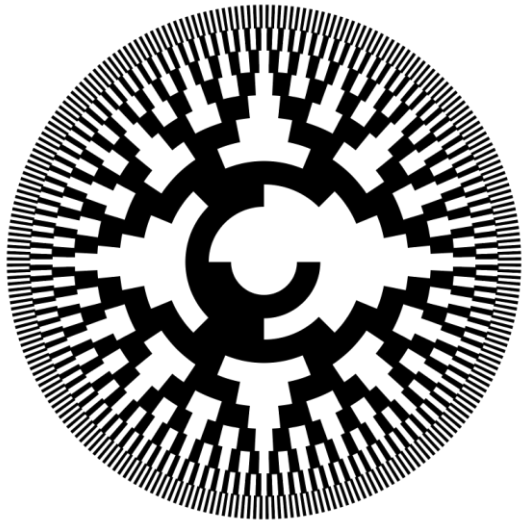
Coding method can lead to robust mixed-signal circuit design.

Gray code was invented by Frank Gray at Bell Lab in 1947.

FRANK GRAY and A. L. Johnsrud in television booth. Behind the glass panels at sides and top are the photo-electric cells.



$Q_2Q_1Q_0$
000
001
011
010
110
111
101
100



Thank you