EE288 Data Conversions/Analog Mixed-Signal ICs
Spring 2018

Lecture 8: Flash ADC

Prof. Sang-Soo Lee
sang-soo.lee@sjsu.edu
ENG-259

Course Schedule – Subject to Change

Date	Topics
24-Jan	Course introduction and ADC architectures
29-Jan	Converter basics: AAF, Sampling, Quantization, Reconstruction
31-Jan	ADC dynamic performance metrics, Spectrum analysis using FFT
5-Feb	ADC & DAC static performance metrics, INL and DNL
7-Feb	OPAMP and bias circuits review
12-Feb	SC circuits review
14-Feb	Sample and Hold Amplifier - Reading materials
19-Feb	Flash ADC and Comparators: Regenerative Latch
21-Feb	Comparators: Latch offset, preamp, auto-zero
26-Feb	Finish Flash ADC
28-Feb	DAC Architectures - Resistor, R-2R
5-Mar	DAC Architectures - Current steering, Segmented
7-Mar	DAC Architectures - Capacitor-based
12-Mar	SAR ADC with bottom plate sampling
14-Mar	SAR ADC with top plate sampling
19-Mar	Midterm Review
21-Mar	Midterm exam
26-Mar	Spring break
28-Mar	Spring break
2-Apr	Pipelined ADC stage - comparator, MDAC, x2 gain
4-Apr	Pipelined ADC bit sync and alignment using Full adders
9-Apr	Pipelined ADC 1.5bit vs multi-bit structures
11-Apr	Fully-differential OPAMP and Switched-capacitor CMFB
16-Apr	Single-slope ADC
18-Apr	Oversampling & Delta-Sigma ADCs
23-Apr	Second- and higher-order Delta-Sigma Modulator.
25-Apr	Hybrid ADC - Pipelined SAR
30-Apr	Hybrid ADC - Time-Interleaving
2-May	ADC testing and FoM
7-May	Project presentation 1
8-May	Project presentation 2
14-May	Final Review
20-May	Project Report Due by 6 PM

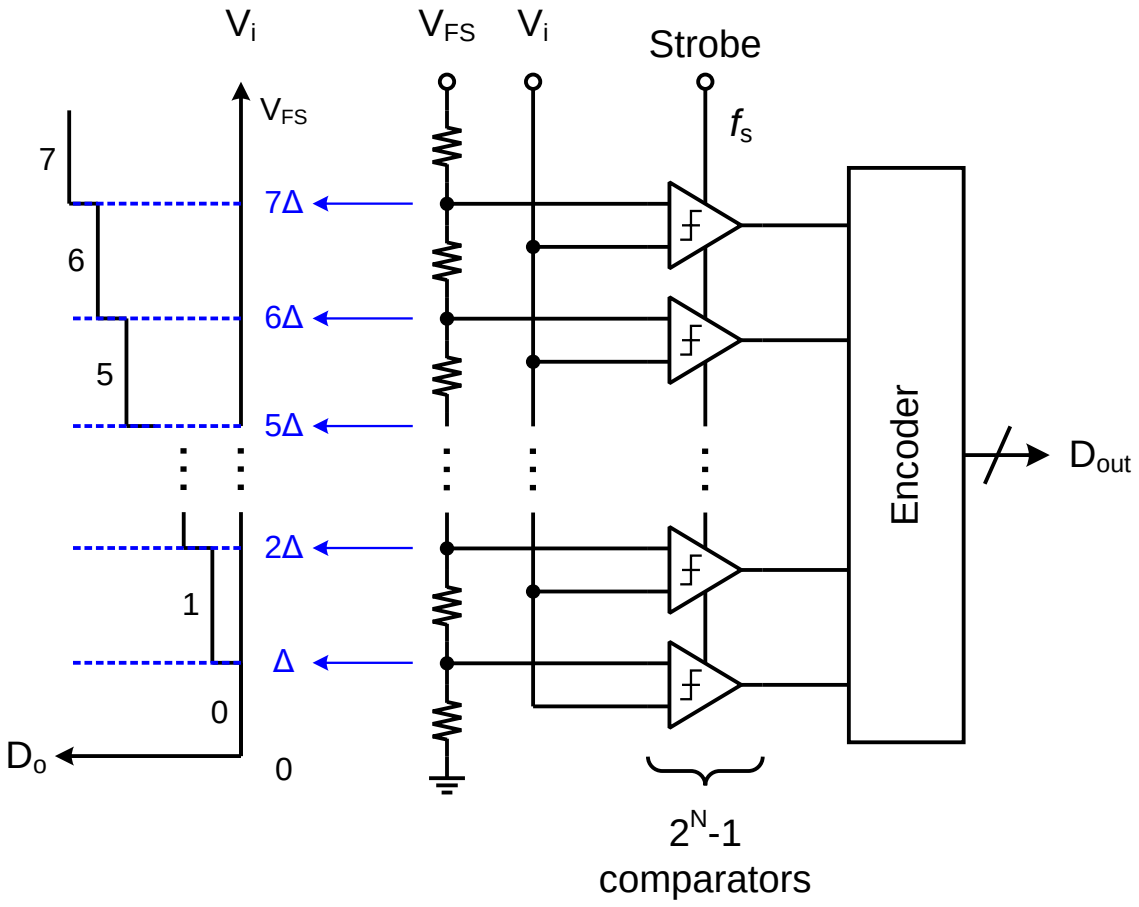
Flash ADC

***Midterm Exam dates are approximate and subject to change with reasonable notice.**

Agenda

- Flash ADC Architecture
- Thermometer to Binary Encoder
- Bubble Issues and Correction Method
- Metastability
- Gray Encoder

Flash ADC



- Reference ladder consists of 2^N equal size resistors
- Input is compared to 2^N-1 reference voltages
- Massive parallelism
- Very fast ADC architecture
- Throughput = f_s
- Latency = 1 T = $1/f_s$
- Complexity = 2^N

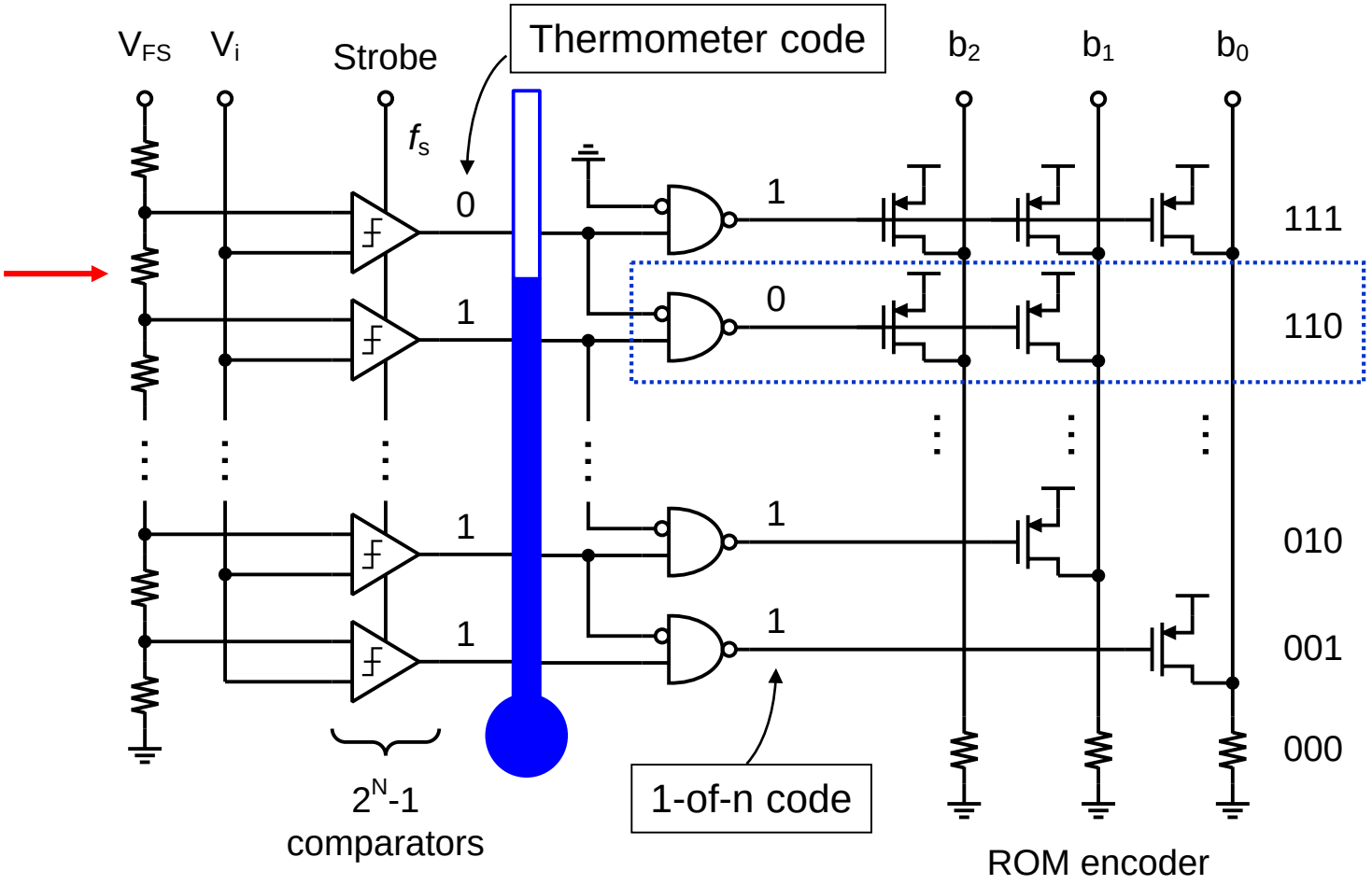
Encoder Implementations

- ROM Encoder – Slow and large power
- Fat Tree Encoder
- Wallace Tree Encoder – Try this if you can!
- Multiplexer Based Encoder

- Reference:

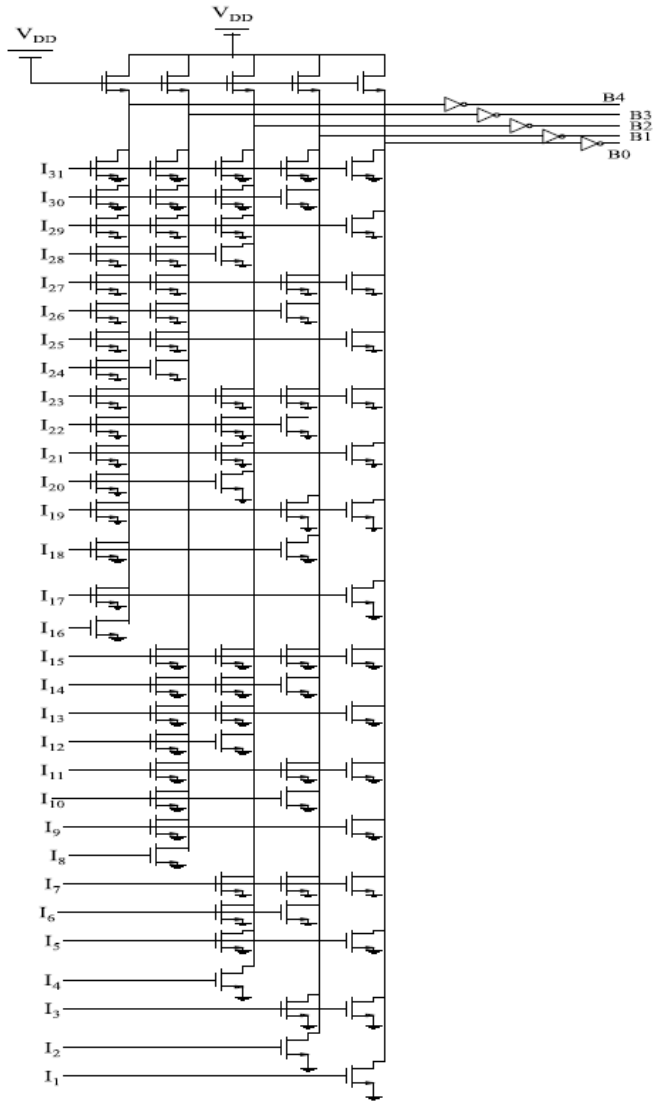
  Chapter 4, “Design and Implementation of a novel flash ADC for ultra wide band applications” PhD Thesis, 2014, GEORGE TOM VARGHESE

Thermometer Code & ROM Encoder



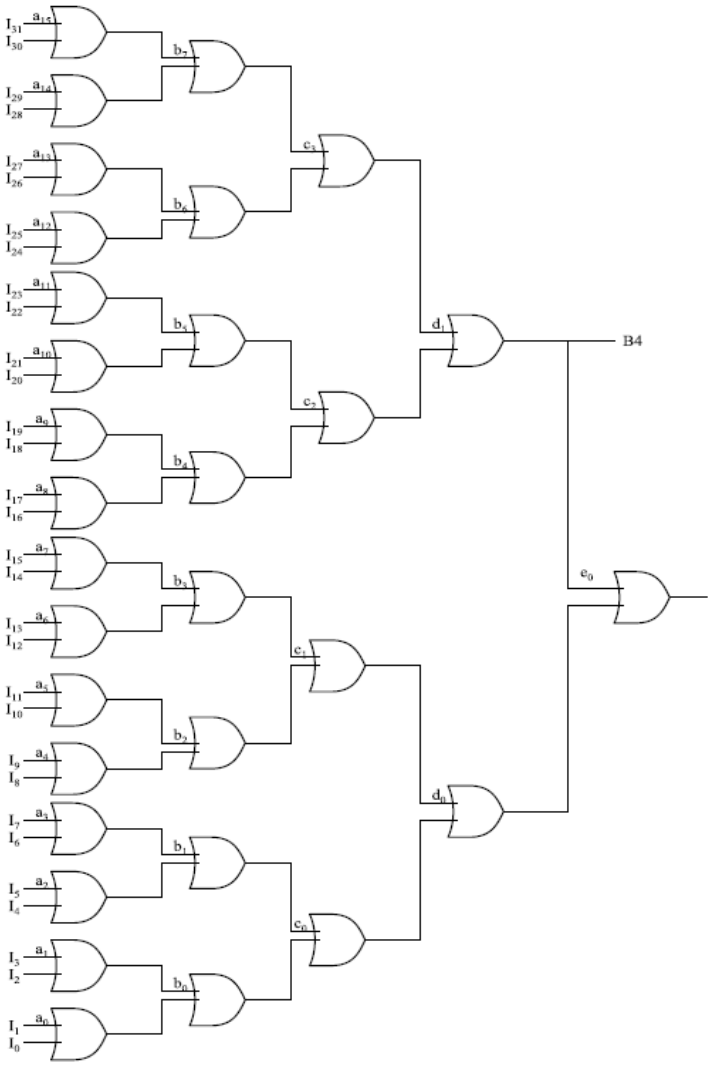
ROM Encoder

1-of-n code



Fat Tree Encoder

1-of-n code



$$B_4 = e_0$$

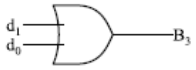
$$B_3 = d_1 + d_0$$

$$B_2 = c_3 + c_2 + c_1 + c_0$$

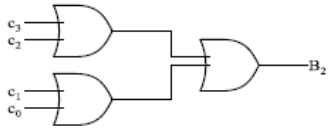
$$B_1 = b_7 + b_6 + b_5 + b_4 + b_3 + b_2 + b_1 + b_0$$

$$B_0 = a_{15} + a_{14} + a_{13} + a_{12} + a_{11} + a_{10} + a_9 + a_8 + a_7 + a_6 + a_5 + a_4 + a_3 + a_2 + a_1 + a_0$$

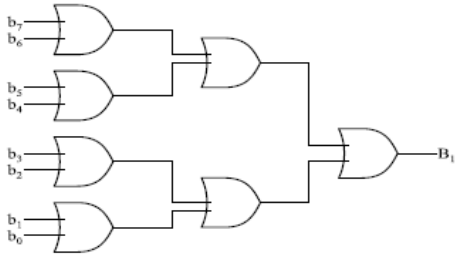
Fat Tree Encoder



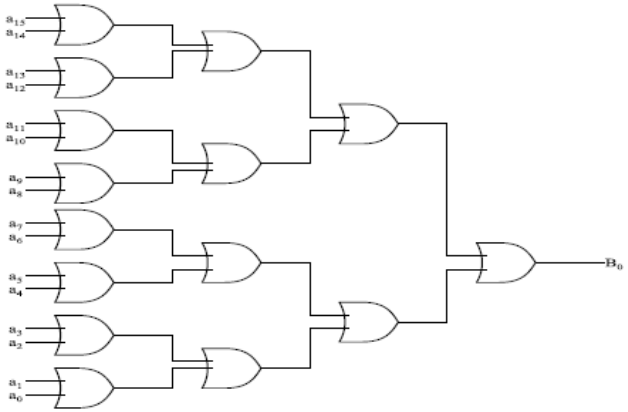
(b) Binary code B_3 generation circuit



(c) Binary code B_2 generation circuit



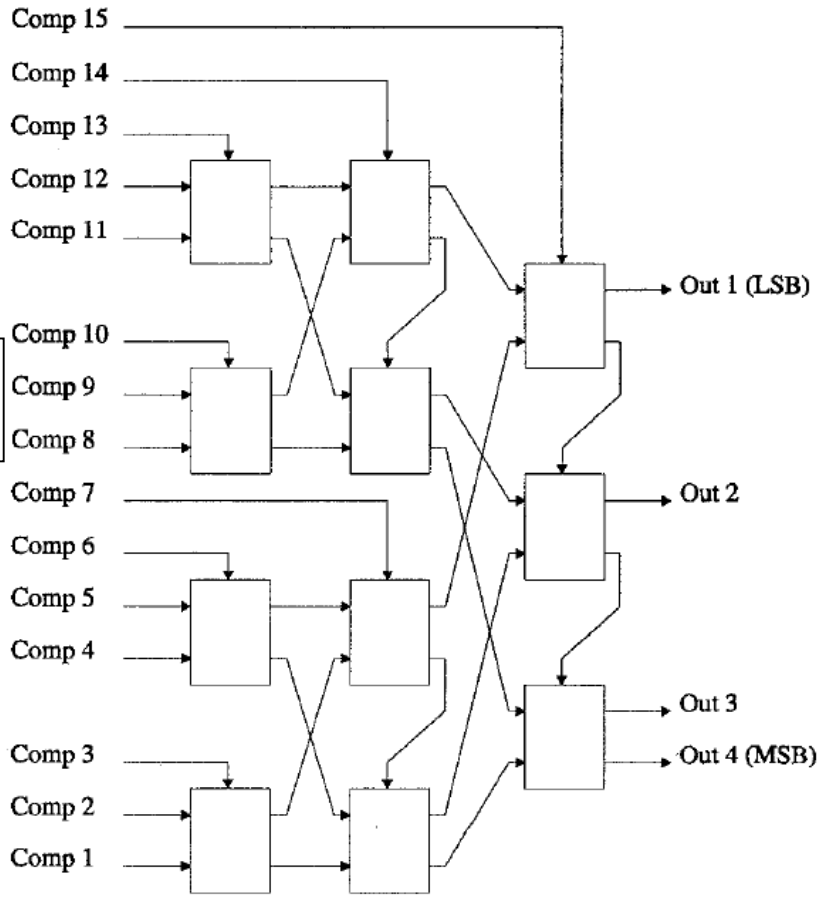
(d) Binary code B_1 generation circuit



(d) Binary code B_0 generation circuit

Wallace Tree Encoder

Thermometer code



Adder Input & outout

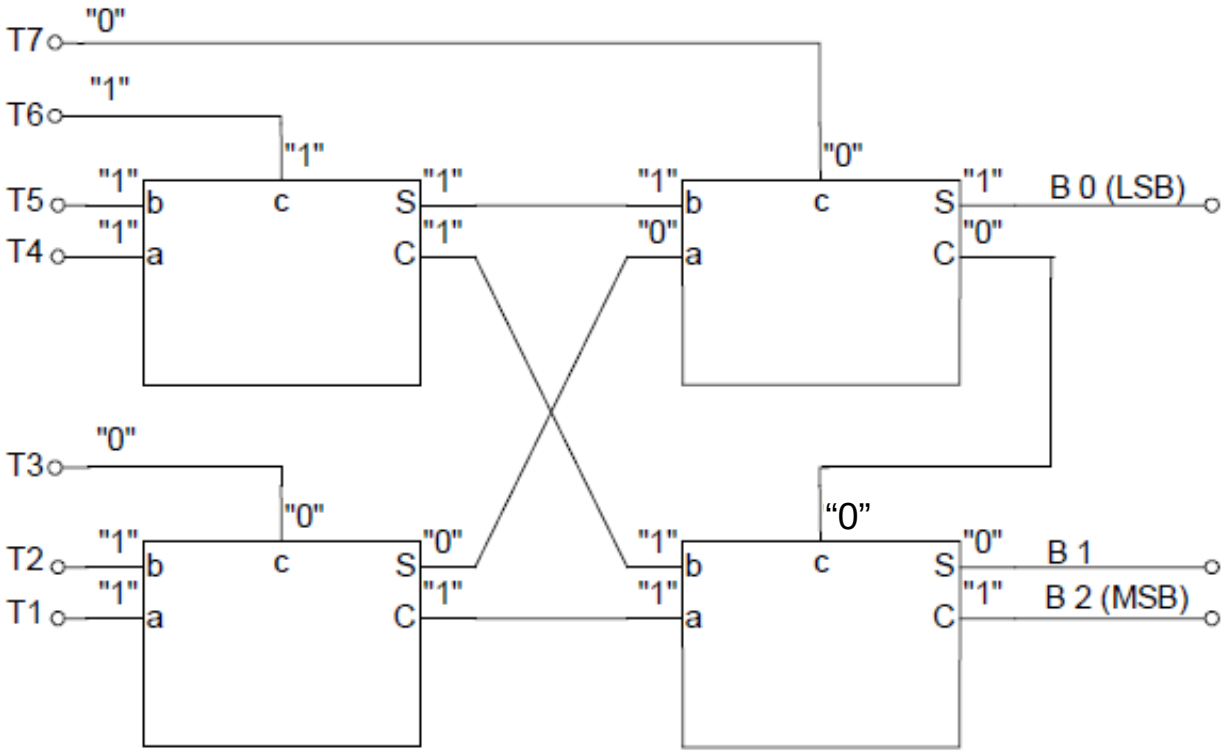
Number of ONE at the entries	Output binary code
0	00
1	01
2	10
3	11

Total number of Adder
$2^N - N - 1$

F. Kaess, R. Kanan, B. Hochet, and M. Declercq, "New encoding scheme for high-speed flash ADC"s," IEEE International Symposium on Circuits and Systems, vol. 1, pp. 5–8, June 1997

Wallace Tree Encoder (3-Bit Example)

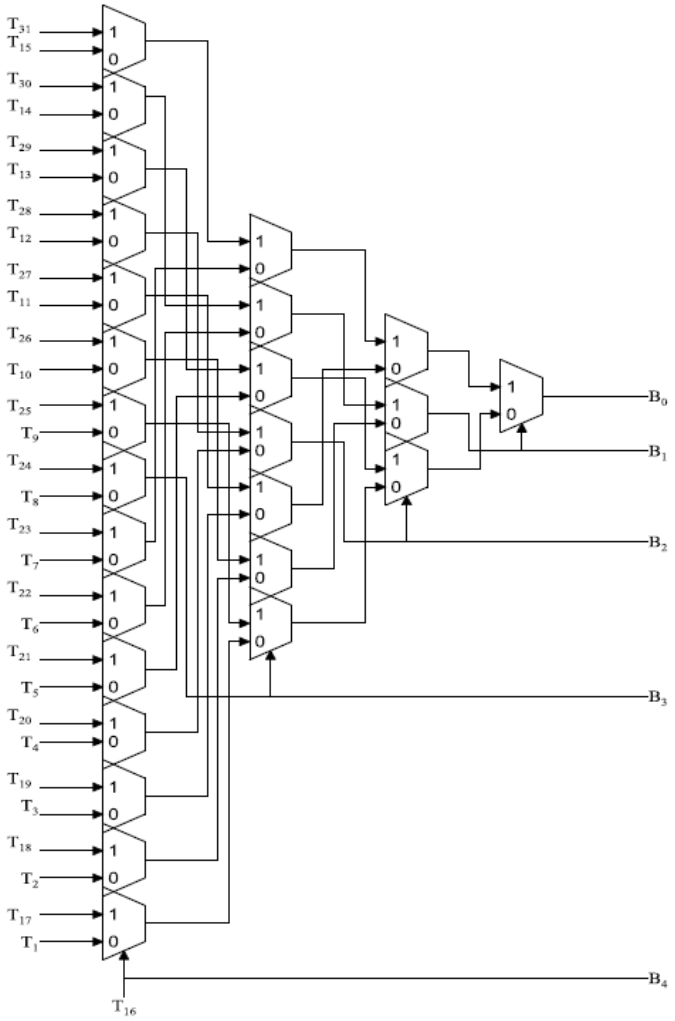
Thermometer code



abc	000	001	010	011	100	101	110	111
CS	00	01	01	10	01	10	10	11

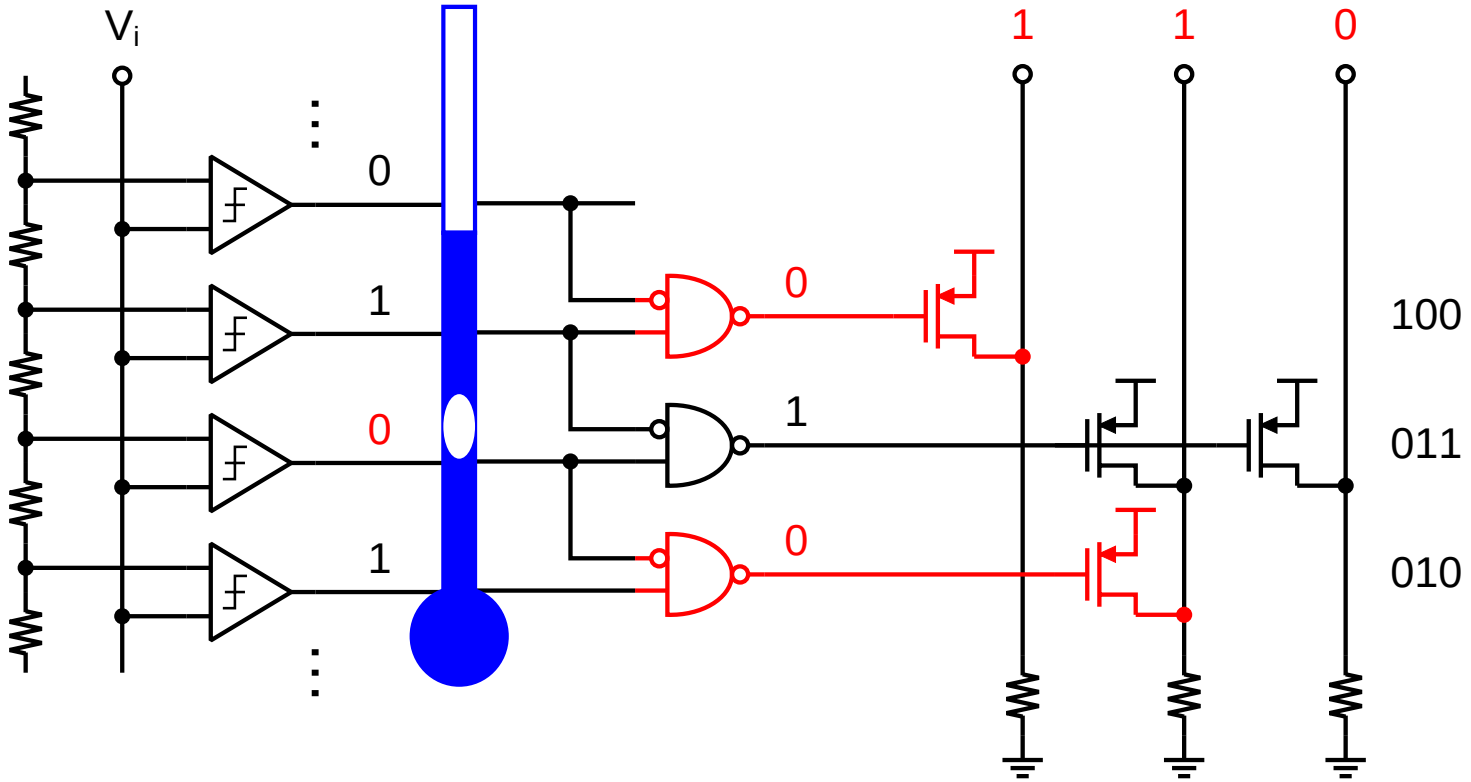
Multiplexer Based Encoder

Thermometer code



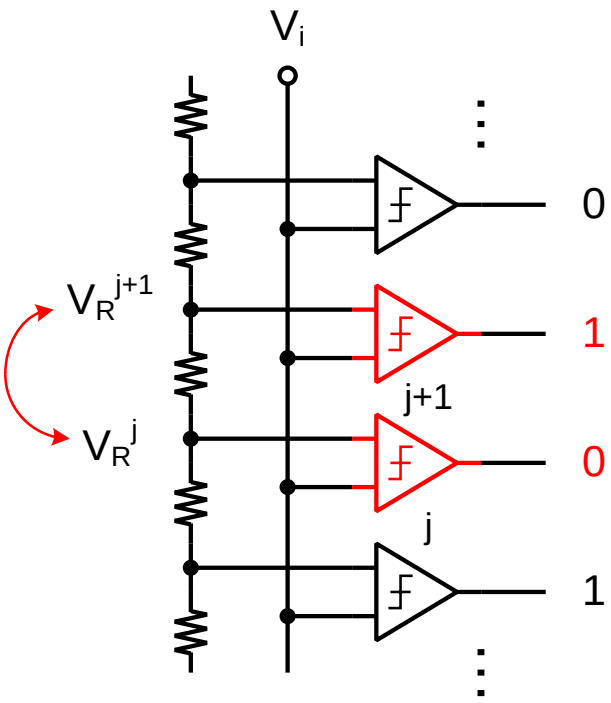
T	B4	B3	B2	B1	B0
31					
30					
29					
28					
27					
26					
25					
24					
23					
22					
21					
20					
19					
18					
17					
16					
15					
14					
13					
12					
11					
10					
9					
8					
7					
6					
5					
4					
3					
2					
1					
0					

Bubbles or Sparkles

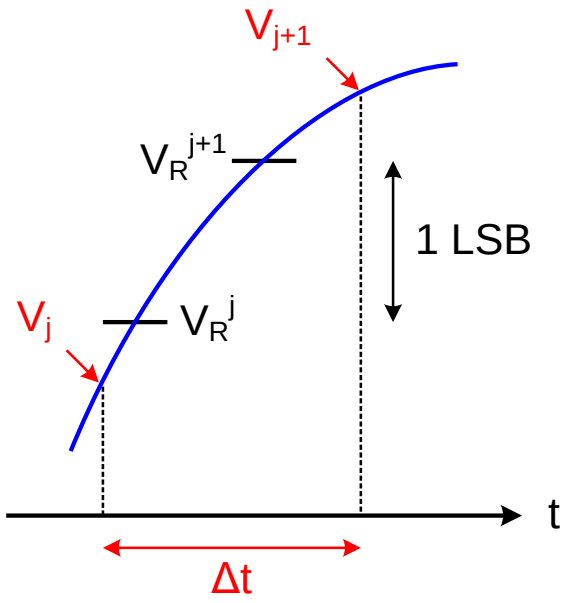


Static/dynamic comparator errors cause bubbles in thermometer code

Cause of Bubbles

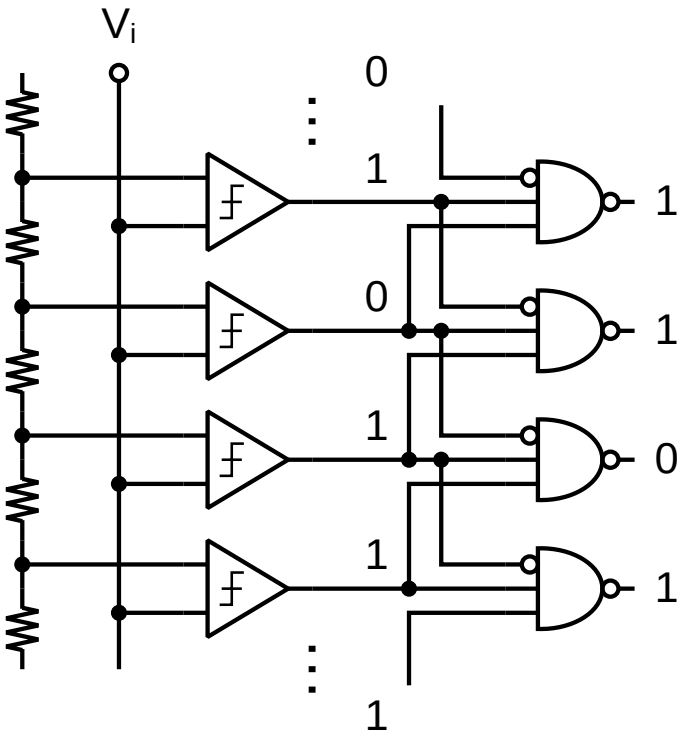


Comparator offset



Timing error

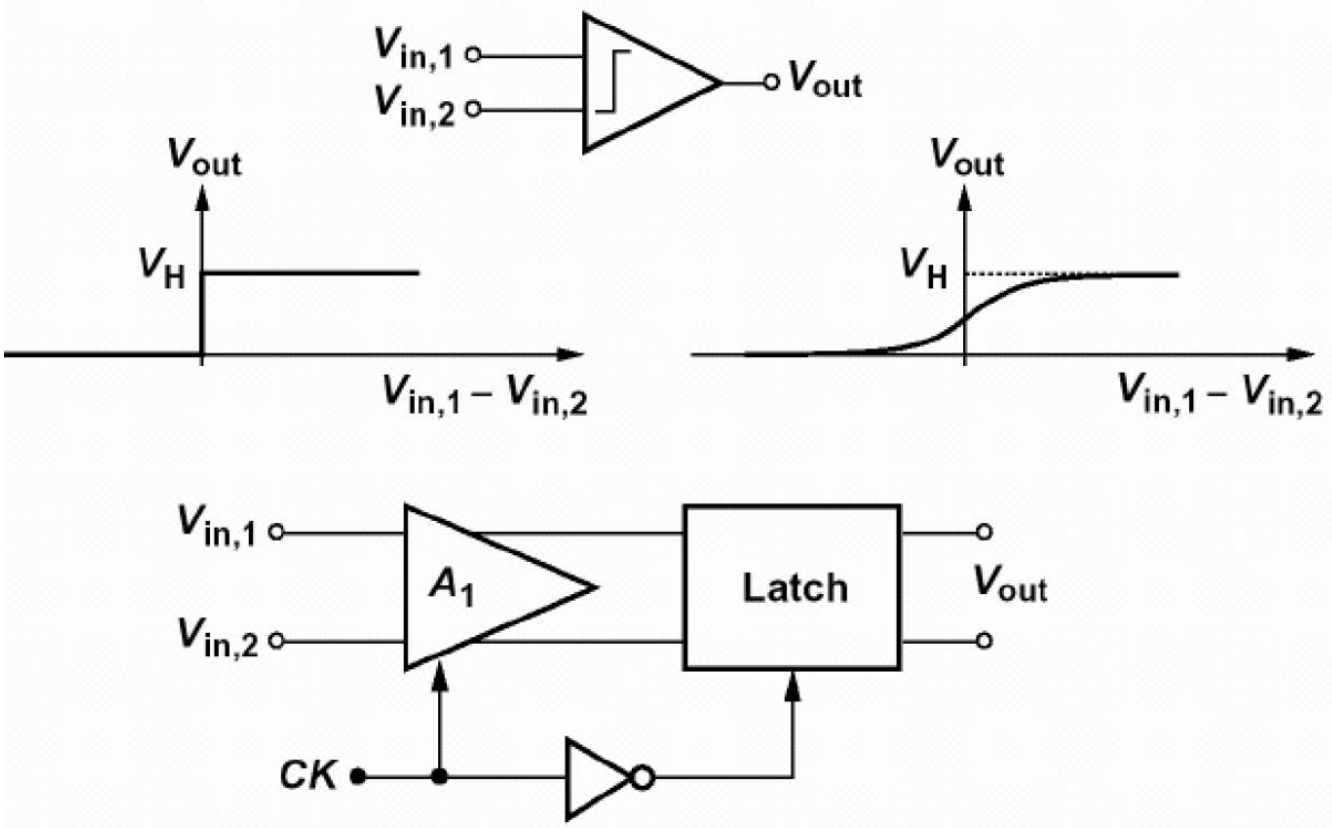
Bubble Correction using 3-input NAND



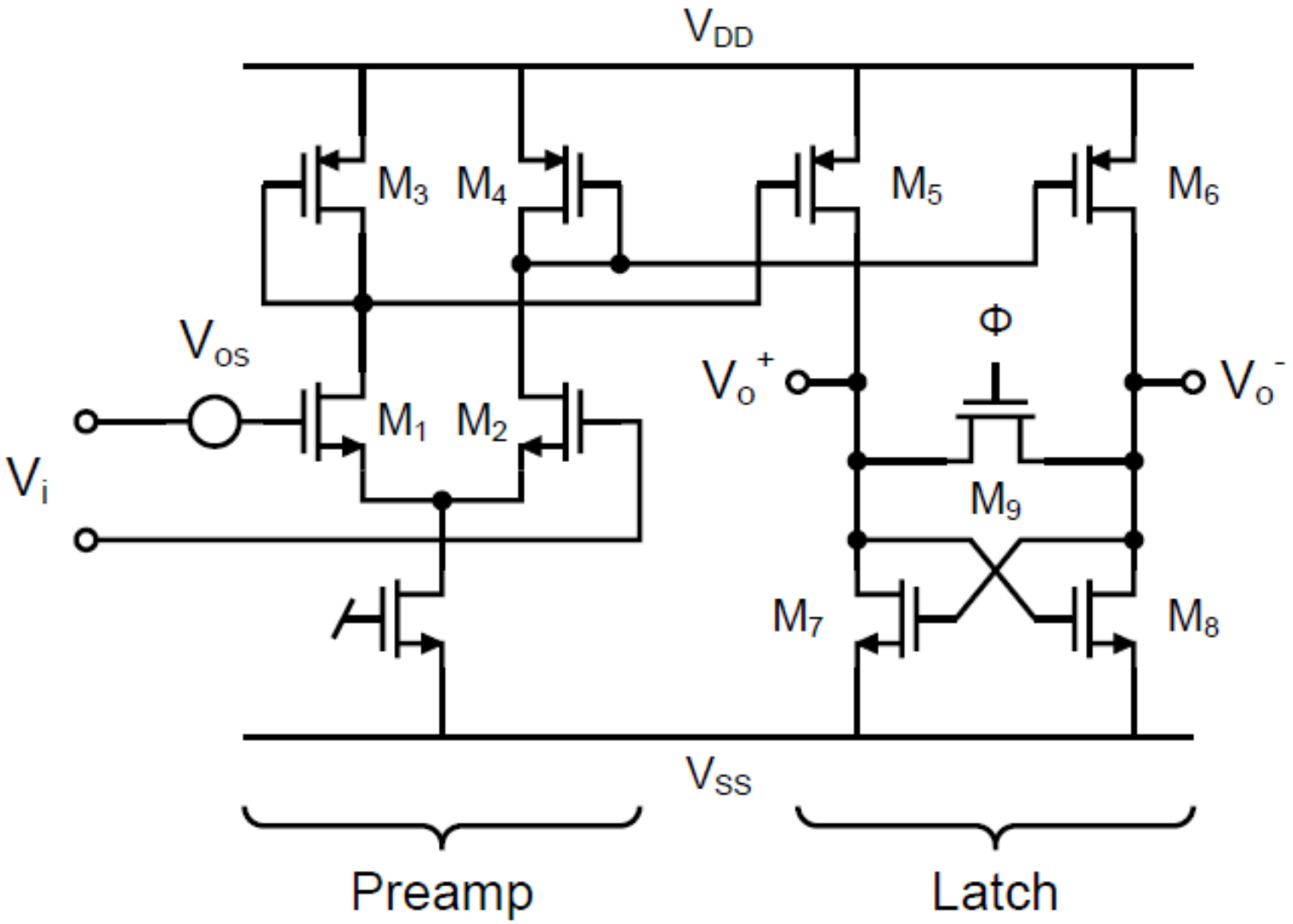
- 3-input NAND
- Detect “011” instead of “01” only

Ref: J. G. Peterson, “A monolithic video A/D converter,” *IEEE Journal of Solid-State Circuits*, vol. 14, pp. 932-937, issue 6, 1979.

Clocked Comparator

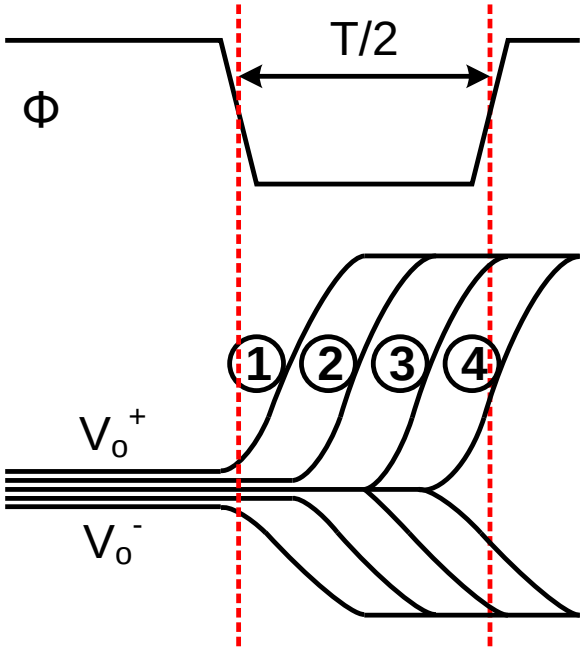


Clocked Comparator Example



Comparator Metastability

If the input difference is small, the outputs take a long time to reach logical levels.

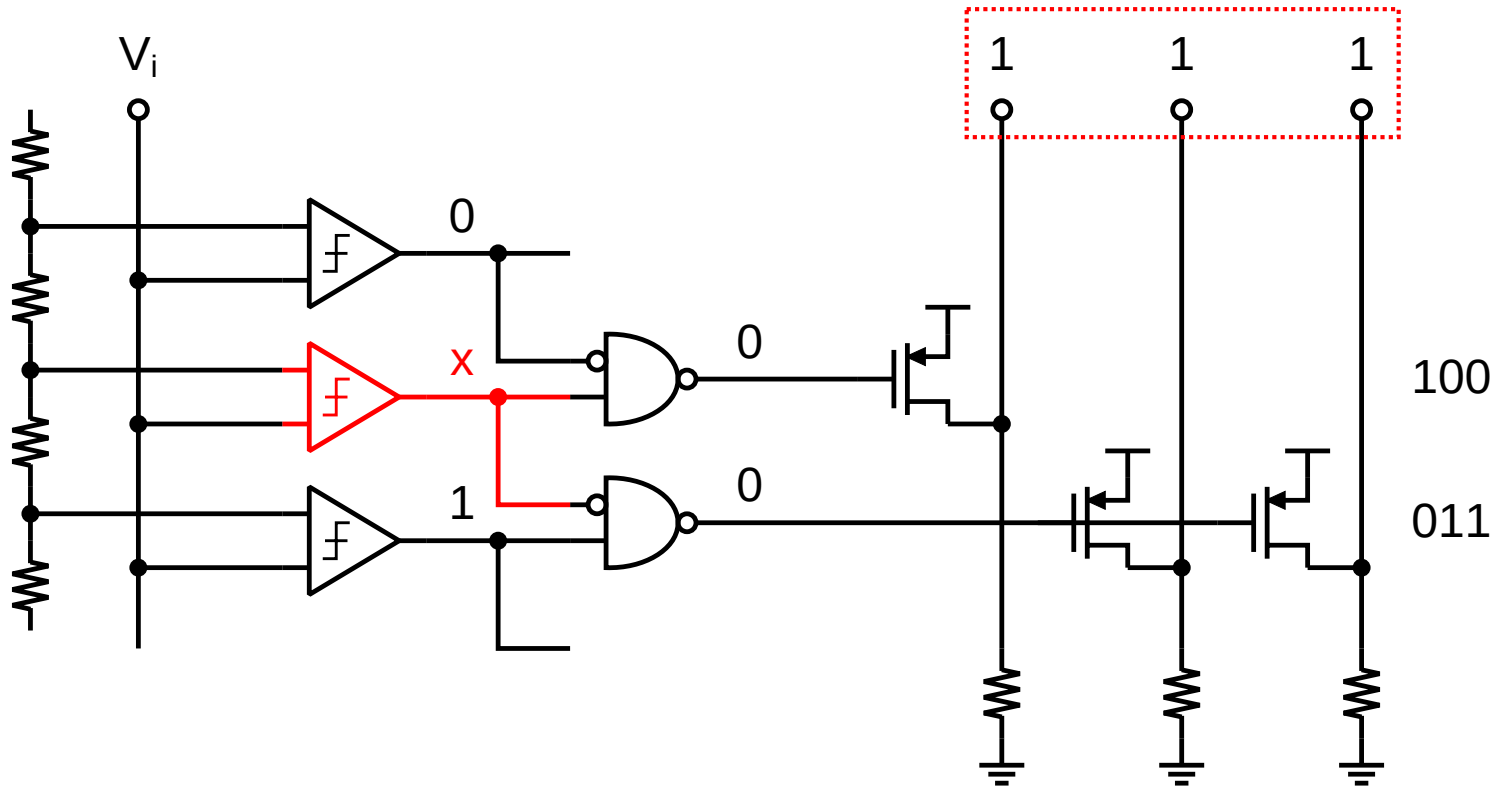


$$V_o(t) = V_i(0) \cdot A_{V1}A_{V2} \cdot \exp(t \cdot g_m/C_L)$$

Curve	$A_{V1}A_{V2}$	V_i(t=0)
①	10	10 mV
②	10	1 mV
③	10	100 μV
④	10	10 μV

Comparator fails to produce valid logic outputs within T/2 when input falls into a region that is sufficiently close to the comparator threshold

Consequence of Comparator Metastability



Logic levels can be misinterpreted by digital gates (branching off, diff. outputs)

– even a wrong decision is better than no decision!

Gray Encoding

$$G_1 = T_1\overline{T_3} + T_5\overline{T_7}$$

$$G_2 = T_2\overline{T_6}$$

$$G_3 = T_4$$

Only one transition b/t adjacent codes

Thermometer							Gray			Binary		
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	1
1	1	0	0	0	0	0	0	1	1	0	1	0
1	1	1	0	0	0	0	0	1	0	0	1	1
1	1	1	1	0	0	0	1	1	0	1	0	0
1	1	1	1	1	0	0	1	1	1	1	0	1
1	1	1	1	1	1	0	1	0	1	1	1	0
1	1	1	1	1	1	1	1	0	0	1	1	1
T_1	T_2	T_3	T_4	T_5	T_6	T_7	G_3	G_2	G_1	B_3	B_2	B_1

- One comparator output is used ONLY once → No branching!
- Gray encoding fails benignly in the presence of bubbles
- Codes are also robust over metastability errors

Gray Encoding

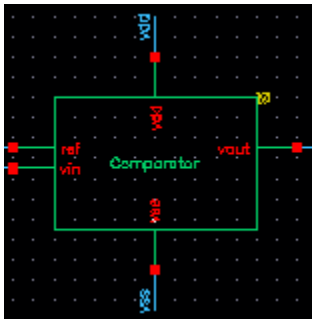
	Thermometer														Gray				Decimal
No Sparkle:	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1	13
One Sparkle:	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	0	15
Two Sparkles:	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1	0	12

$$G_1 = T_1\overline{T_3} + T_5\overline{T_7}$$
$$G_2 = T_2\overline{T_6}$$
$$G_3 = T_4$$

Conversion of Gray code to binary code is quite time-consuming → "quasi" Gray code

Ref: Y. Akazawa, et al., "A 400MSPS 8b flash AD conversion LSI," in *IEEE International Solid-State Circuits Conference*, Dig. Tech. Papers, 1987, pp. 98-99.

Verilog-A code for Comparator




```
// VerilogA for ADC2,Comparator, veriloga

`include "constants.vams"
`include "disciplines.vams"

module Comparator(vin, vout, vdd, vss, ref);

inout vdd, vss;
output vout;
input vin, ref;
electrical vin, ref, vout, vdd, vss;

parameter delay = 0, ttime = 1p;
real result;

 analog begin
           @(cross((V(vin)-V(ref)), 0))
                      if (V(vin) >= V(ref))
                                 result = V(vdd);
                      else
                                 result = V(vss);

           V(vout) <+ transition(result, delay, ttime);

 end

endmodule                          // Comparator
```

Verilog-A Reference

https://verilogams.com/tutorials/vloga-intro.html

http://help.simetrix.co.uk/8.0/simetrix/mergedProjects/verilog_a_reference/topics/veriloga_verilog_areference_verilog_afunctions.htm