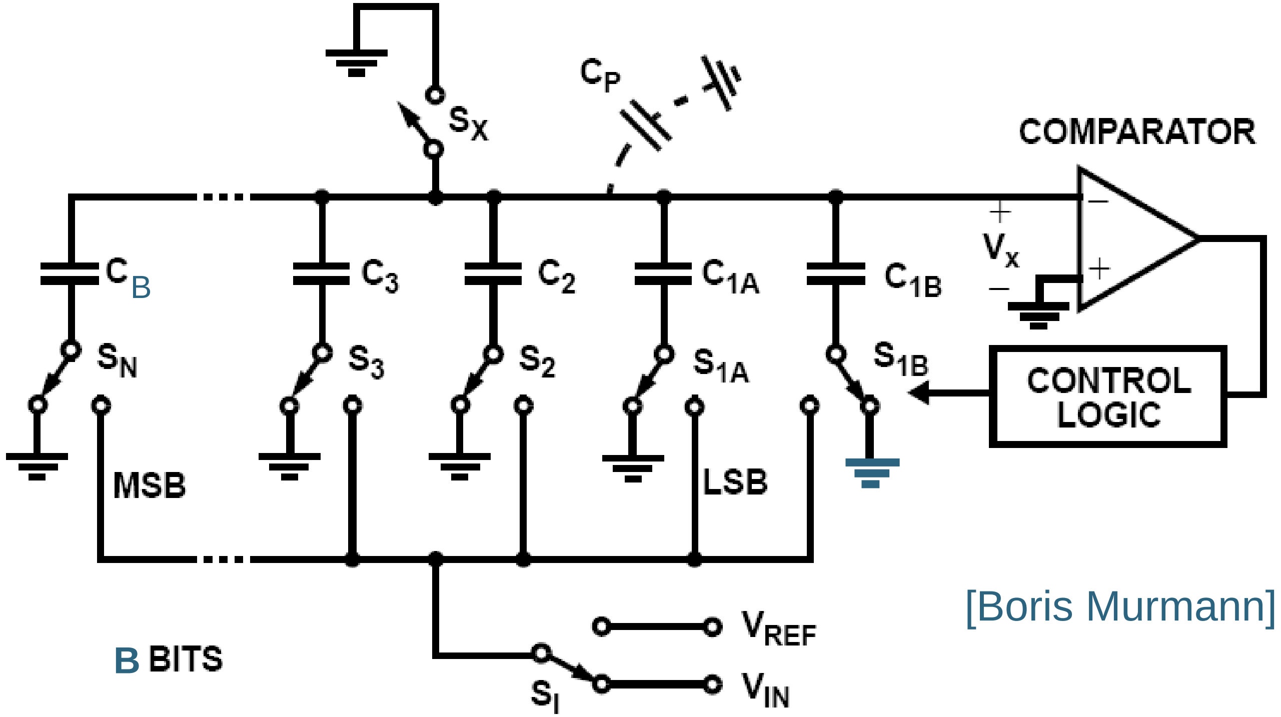
# EE 240C
# Analog-Digital Interface Integrated Circuits

# SAR ADC Capacitor and Switching Schemes

# SAR Techniques

- Small custom unit capacitors for low- to mid-resolution SAR ADC
- DAC schemes and switching schemes for low power or lower area
- Self-timed comparator or asynchronous SAR
  - At most 2 critical comparator decisions
  - Optimize time needed for comparator
- Redundant SAR
  - Make SAR robust against DAC settling errors and comparator noise
- SAR DAC Calibration
  - DAC non-linearity
  - Digital overhead

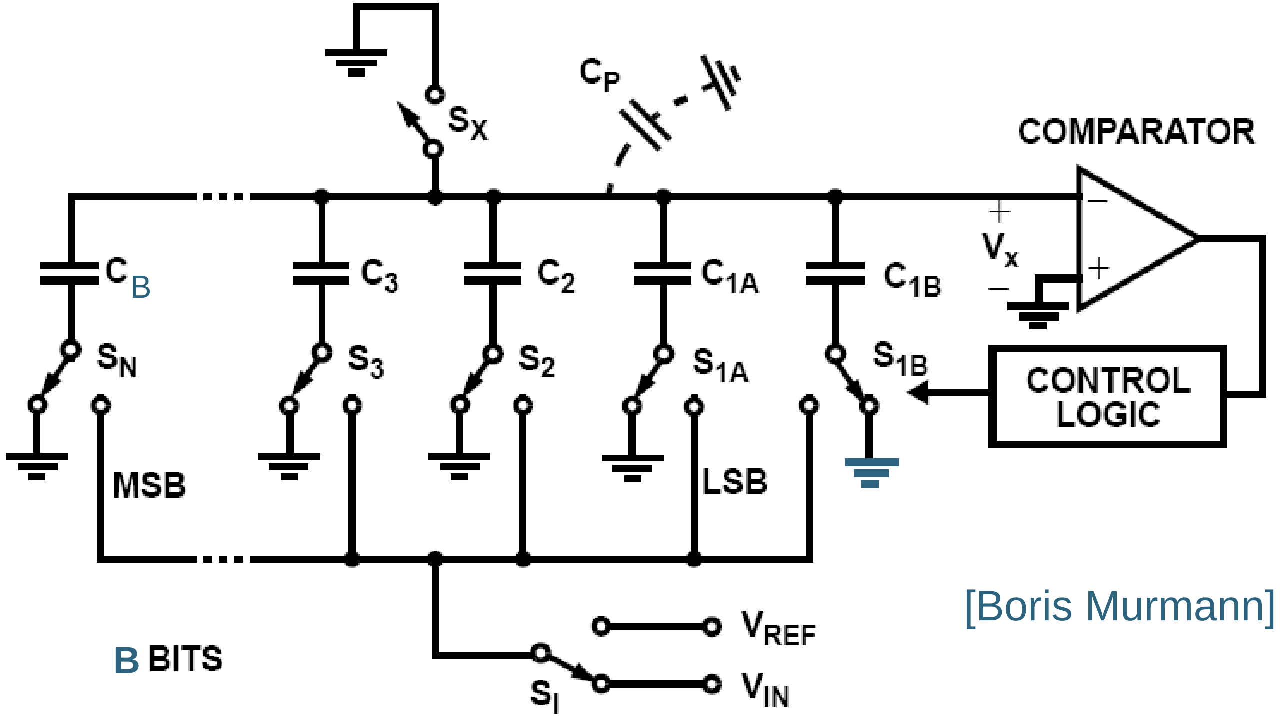
# Conventional Switching Scheme



[Boris Murmann]

- Fully differential variants
- Unsuccessful bit trials lead to reference power loss
- Comparator input node swing

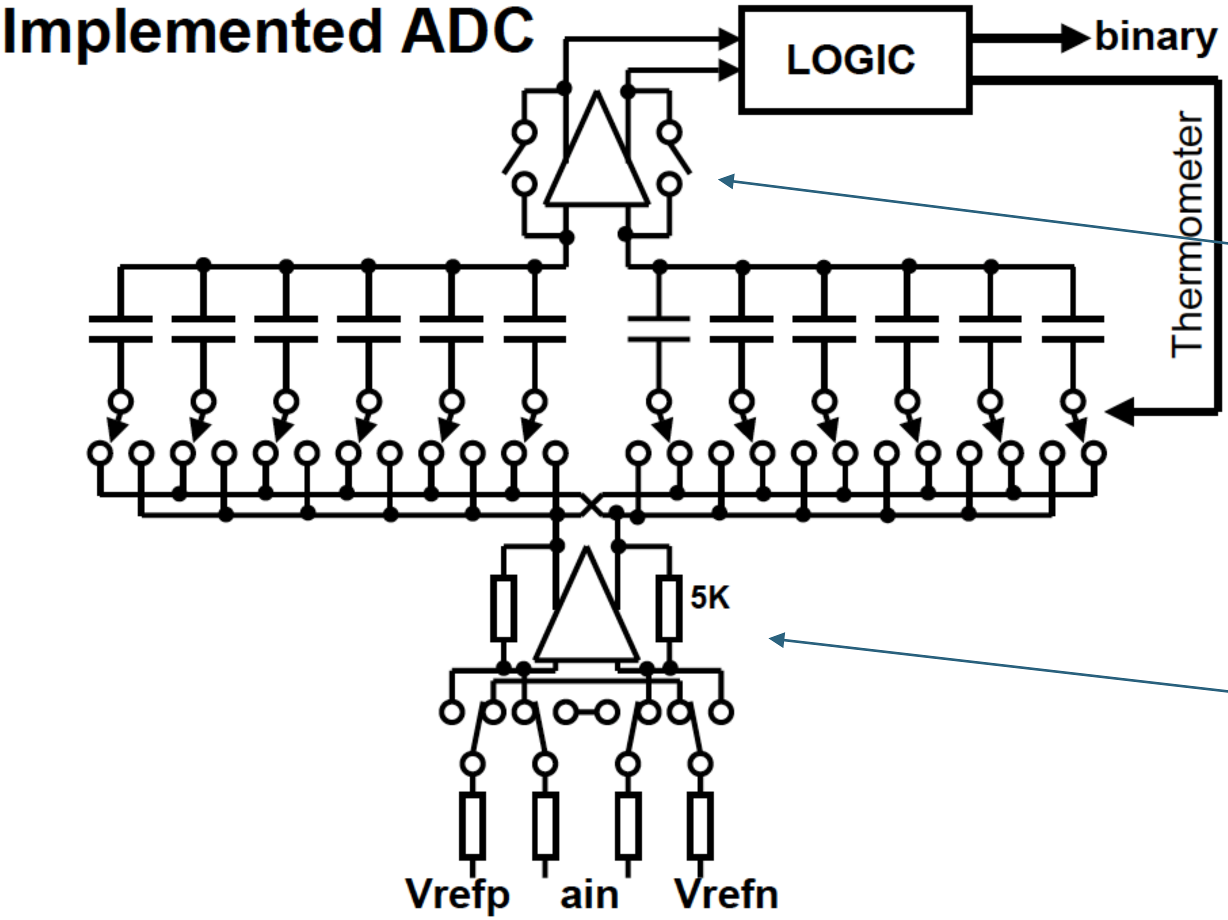
# Conventional Switching Scheme



[Boris Murmann]

- Parasitic capacitor $C_p$ only causes smaller comparator signal
  - Capacitive divider between switched capacitor and parasitic capacitor

# Conventional Switching Scheme



[Kuttner2002]

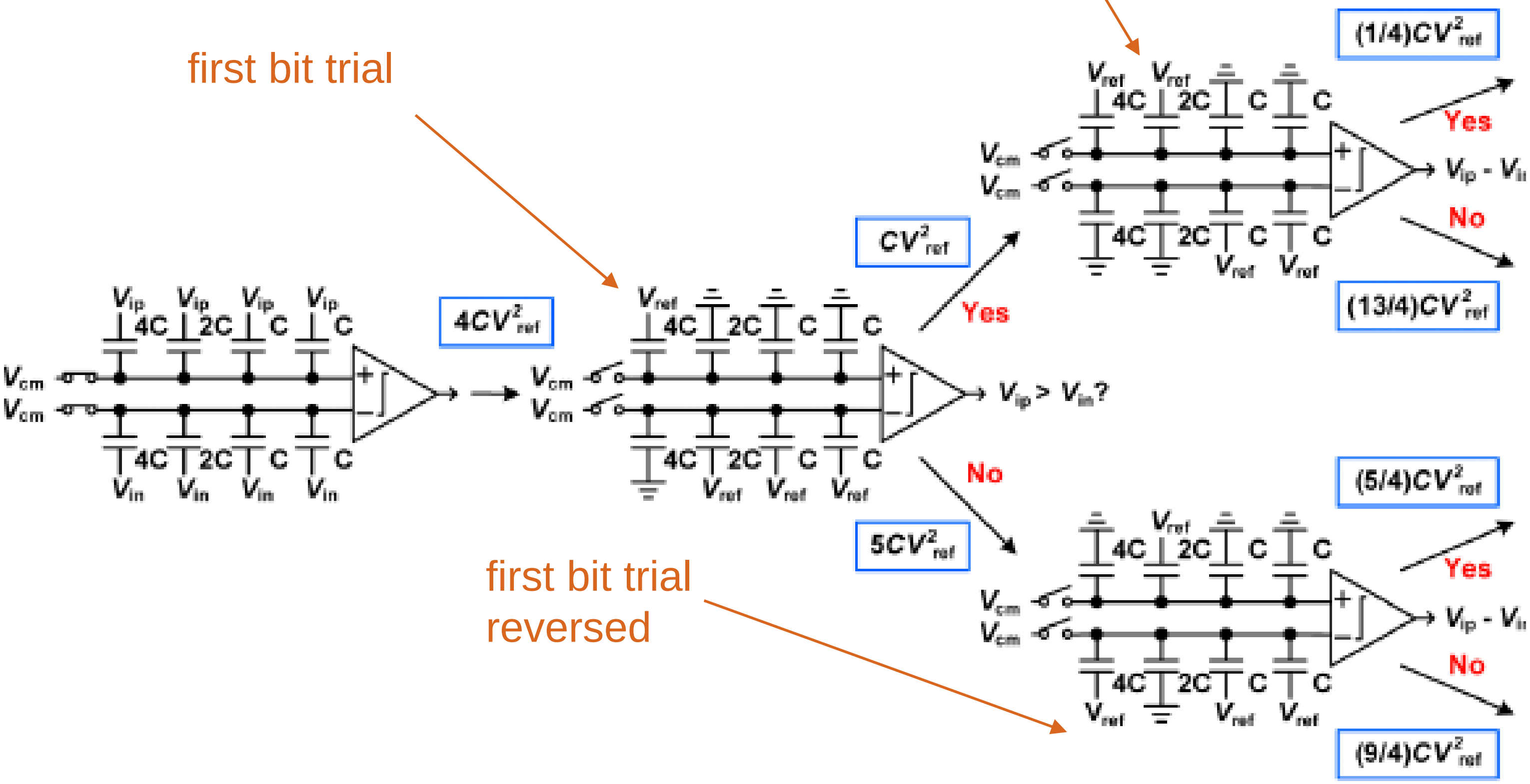
AZ comparator
(input offset compensation)

Input buffer and
reference buffer

- Fully differential implementation

# Conventional Switching Scheme

- Conventional Switching Scheme



second bit trial

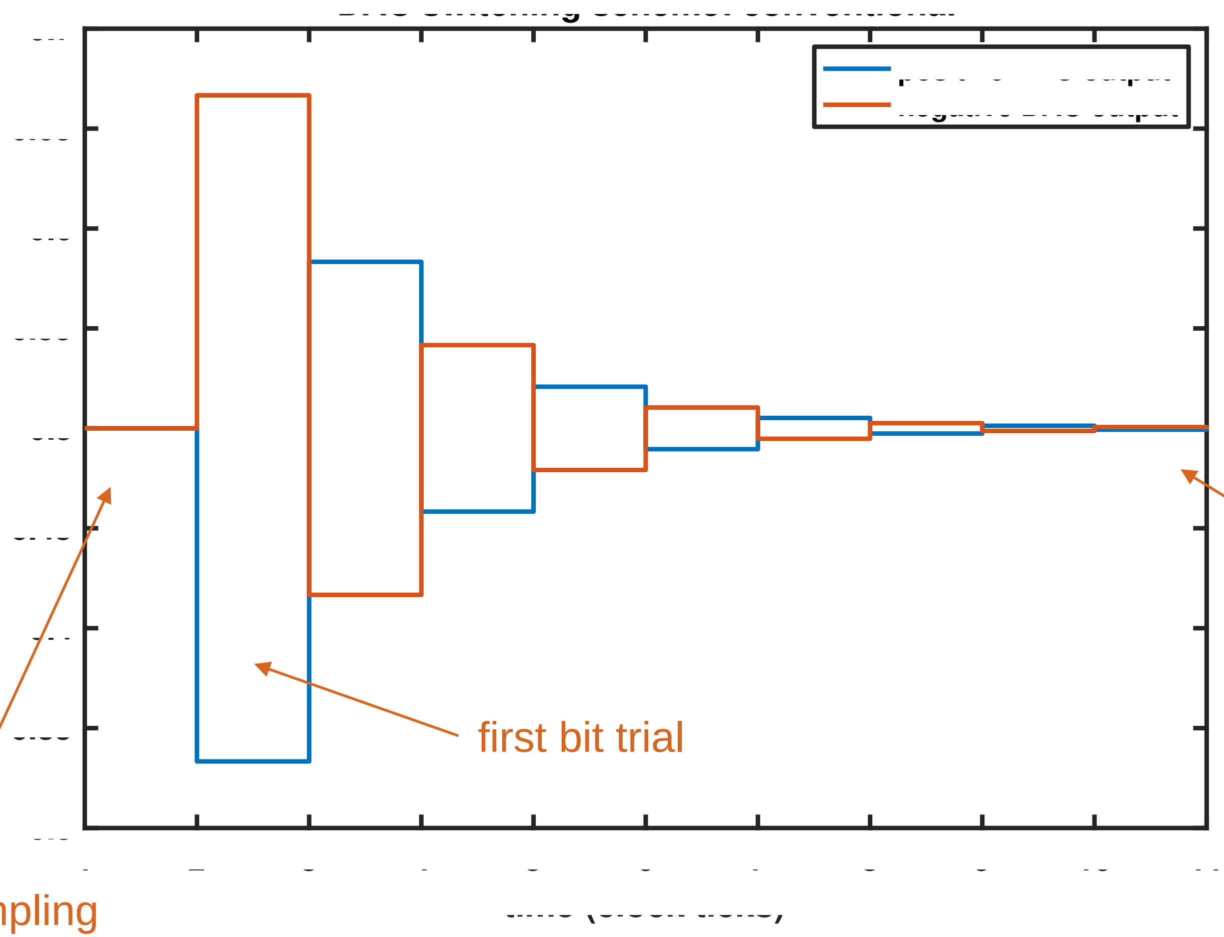
first bit trial

first bit trial reversed

$$Q = -V_{in} \cdot C_{\text{total}} = (V_x - V_{DAC}) \cdot C_{\text{total}} + V_x \cdot C_p$$

# Conventional Switching Scheme

- Comparator input voltage example for a 10-bit ADC



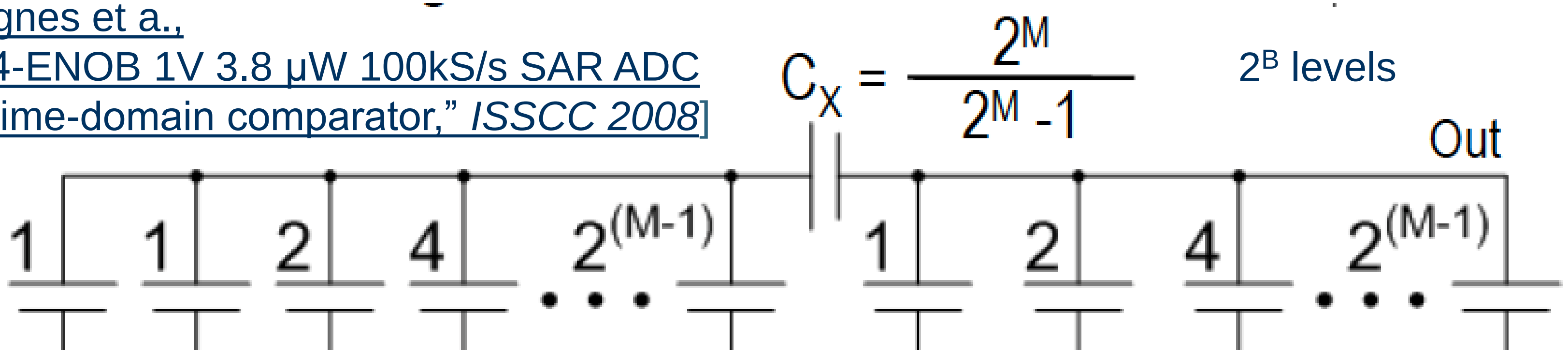
input sampling

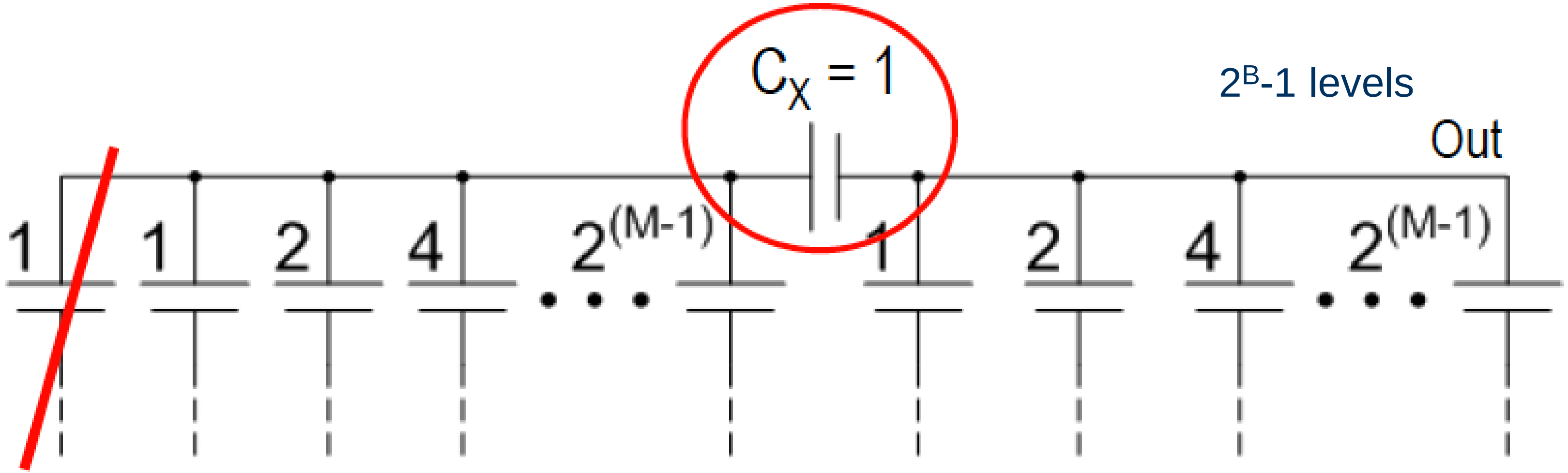
first bit trial

Differential mode
converges to 0;
common mode constant

# Split Capacitor Array

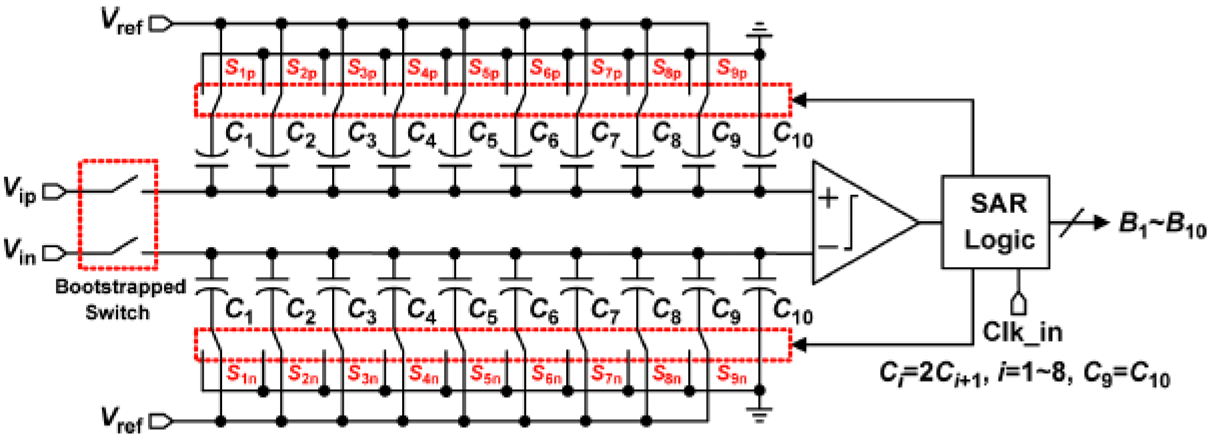
[A. Agnes et a., "A 9.4-ENOB 1V 3.8 μW 100kS/s SAR ADC with time-domain comparator," *ISSCC 2008*]





- Smaller number of unit elements, but needs calibration
  - Sensitive to parasitic capacitor on secondary summing node

# Monotonic Switching Scheme

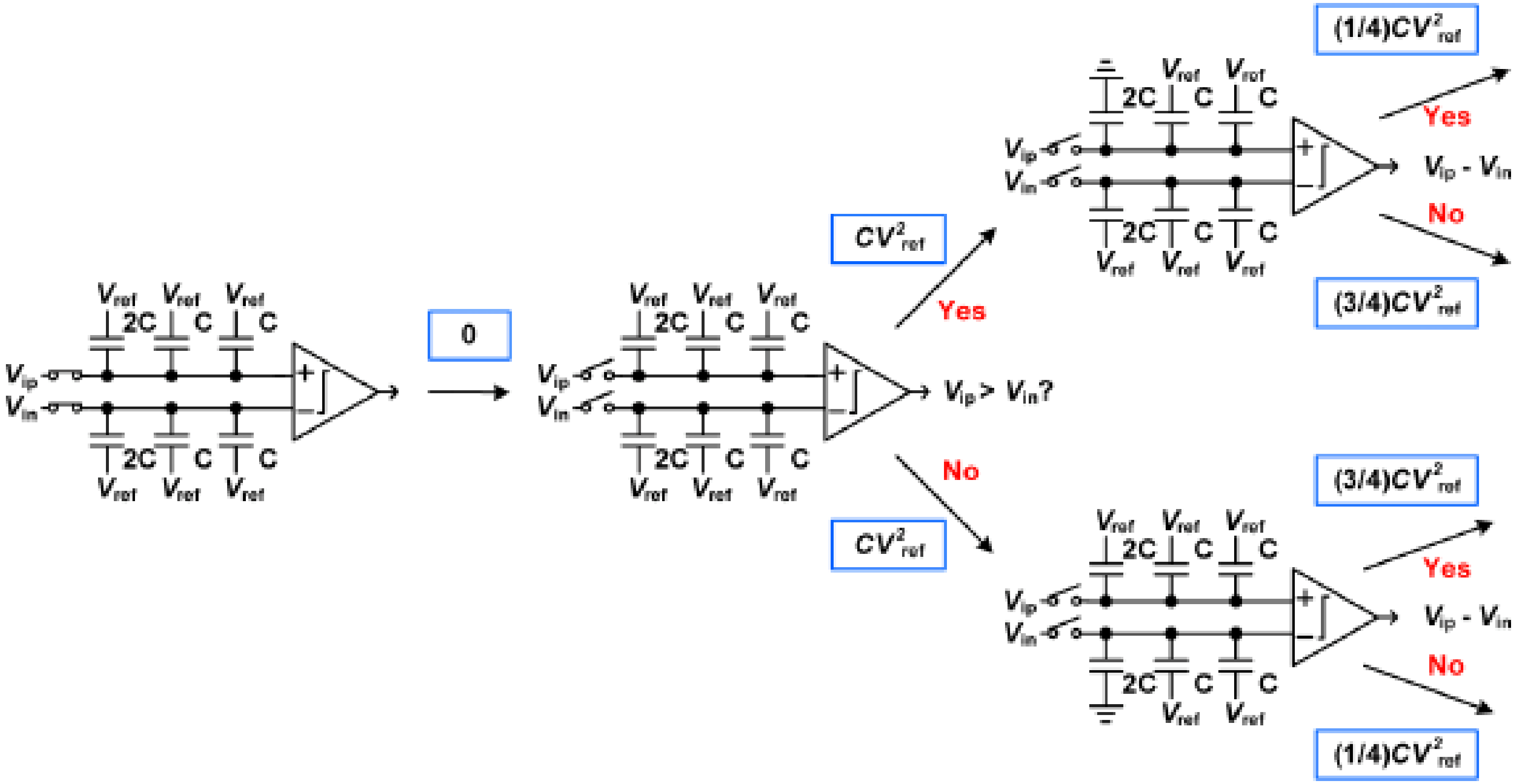

$V_{ref}$
$S_{1p}$ $S_{2p}$ $S_{3p}$ $S_{4p}$ $S_{5p}$ $S_{6p}$ $S_{7p}$ $S_{8p}$ $S_{9p}$
$C_1$ $C_2$ $C_3$ $C_4$ $C_5$ $C_6$ $C_7$ $C_8$ $C_9$ $C_{10}$
$V_{ip}$
$V_{in}$
+
−
SAR Logic
$B_1$~$B_{10}$
Bootstrapped Switch
$C_1$ $C_2$ $C_3$ $C_4$ $C_5$ $C_6$ $C_7$ $C_8$ $C_9$ $C_{10}$
Clk_in
$S_{1n}$ $S_{2n}$ $S_{3n}$ $S_{4n}$ $S_{5n}$ $S_{6n}$ $S_{7n}$ $S_{8n}$ $S_{9n}$
$C_i$=$2C_{i+1}$, $i$=1~8, $C_9$=$C_{10}$
$V_{ref}$

[C.-C. Liu et al., "A 10-bit 50-MS/s SAR ADC with a monotonic capacitor switching procedure," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 731–740, Apr. 2010]

- Top-plate sampling
- Compare before switching (B-1 single-ended switches for B bits)
- Parasitic capacitor $C_p$ (not shown) leads to gain error

# Monotonic Switching Scheme

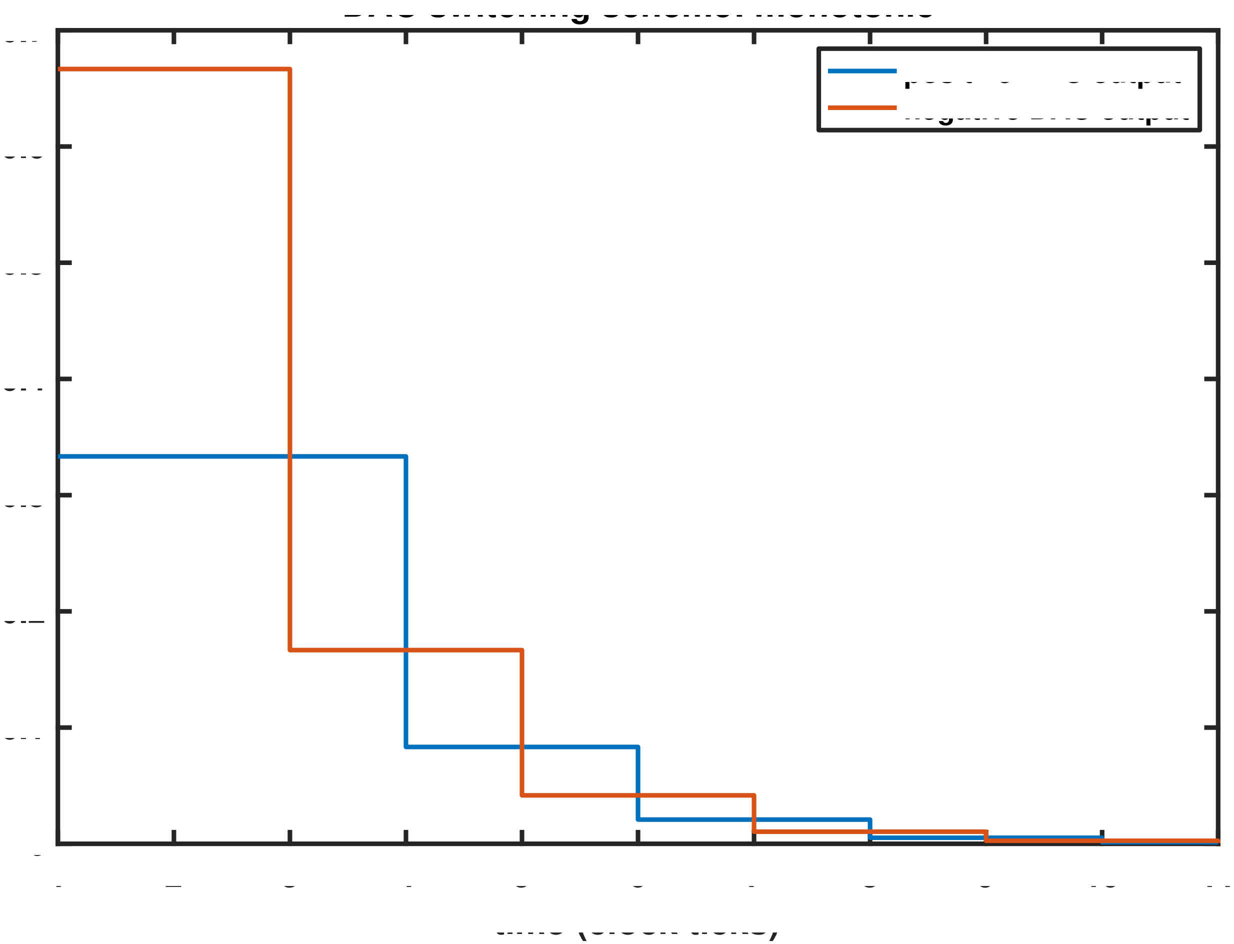
- Monotonic Switching Scheme



$$Q = V_{in} \cdot (C_{\text{total}} + C_p) = (V_x - V_{DAC}) \cdot C_{\text{total}} + V_x \cdot C_p$$

# Monotonic Switching Scheme

- Comparator input voltage example for a 10-bit ADC



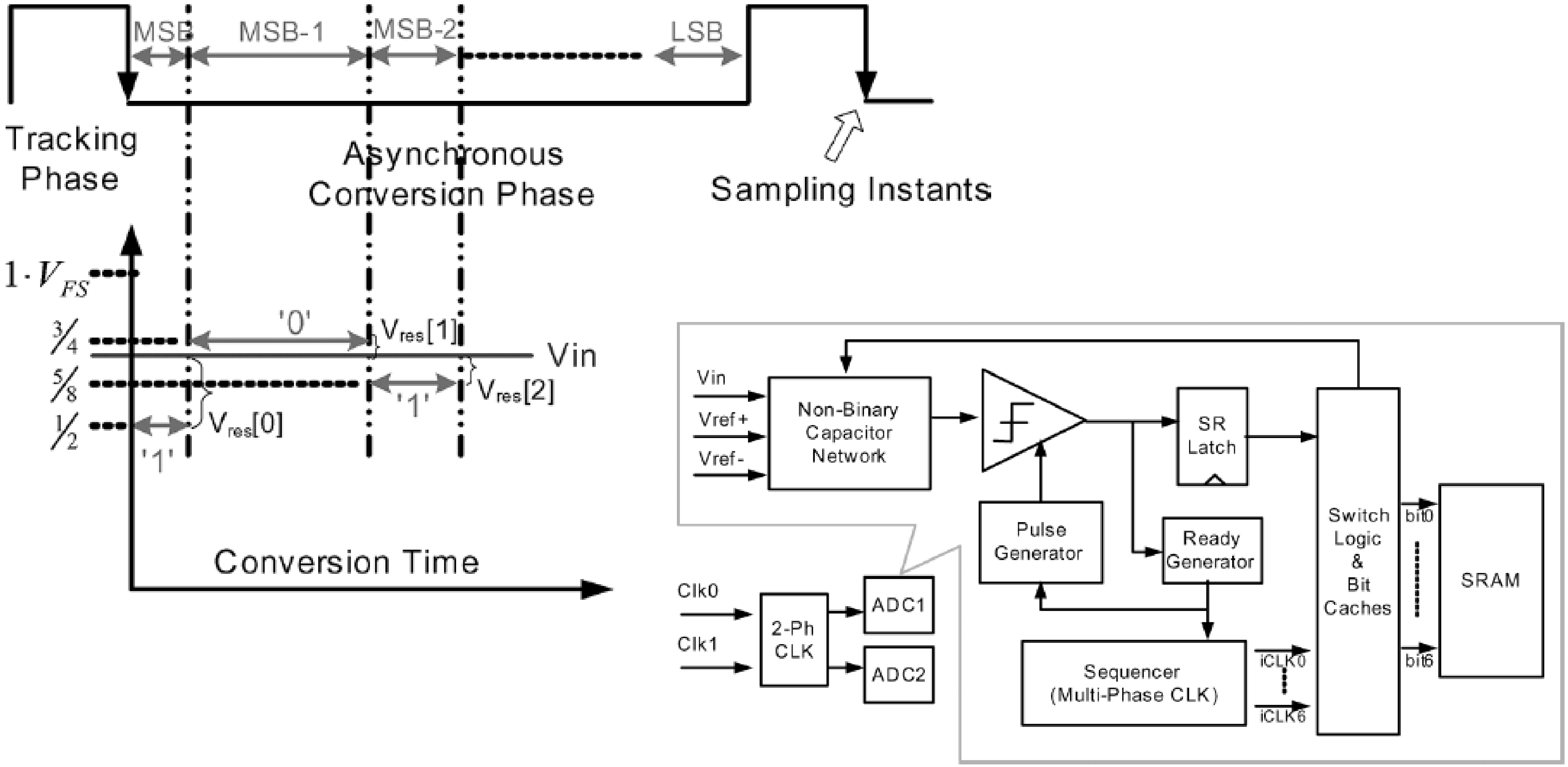
# EE 240C
# Analog-Digital Interface Integrated Circuits

## Asynchronous SAR ADC

# SAR Techniques

- Small custom unit capacitors for low- to mid-resolution SAR ADC
- DAC schemes and switching schemes for low power or lower area
- Self-timed comparator or asynchronous SAR
  - At most 2 critical comparator decisions
  - Optimize time needed for comparator
- Redundant SAR
  - Make SAR robust against DAC settling errors and comparator noise
- SAR DAC Calibration
  - DAC non-linearity
  - Digital overhead

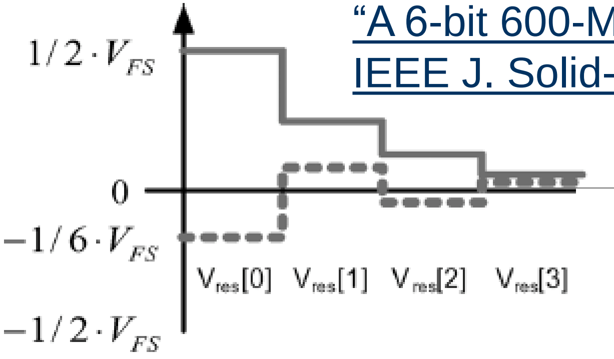
# Asynchrous SAR ADC

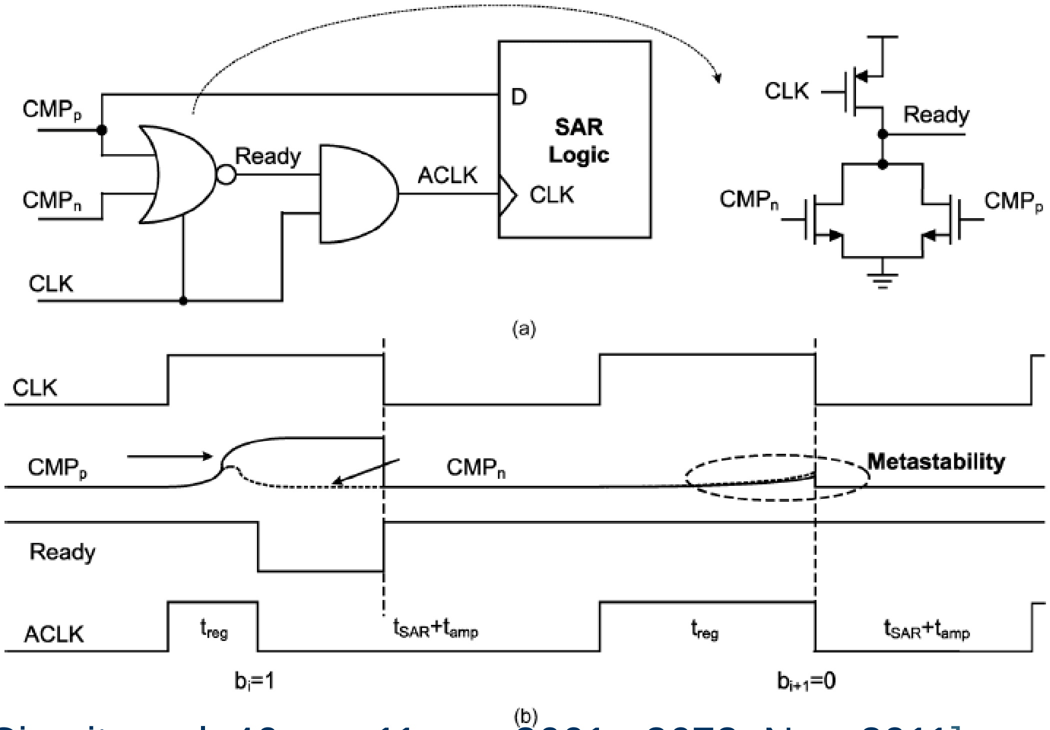


[S. W. M. Chen and R. W. Brodersen,
“A 6-bit 600-MS/s 5.3-mW asynchronous ADC in 0.13- m CMOS,”
IEEE J. Solid-State Circuits, vol. 41, no. 12, pp. 2669–2680, Dec. 2006]

# Asynchronous SAR ADC: Metastability

[S. W. M. Chen and R. W. Brodersen, "A 6-bit 600-MS/s 5.3-mW asynchronous ADC in 0.13- m CMOS," IEEE J. Solid-State Circuits, vol. 41, no. 12, pp. 2669–2680, Dec. 2006]



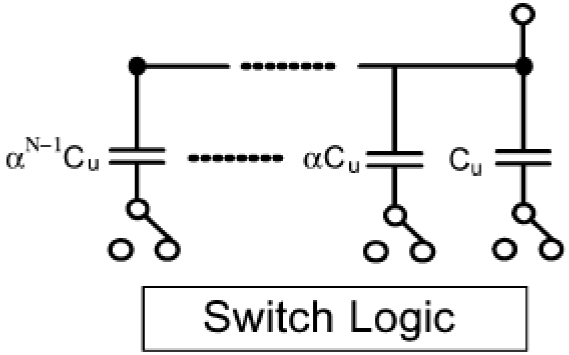


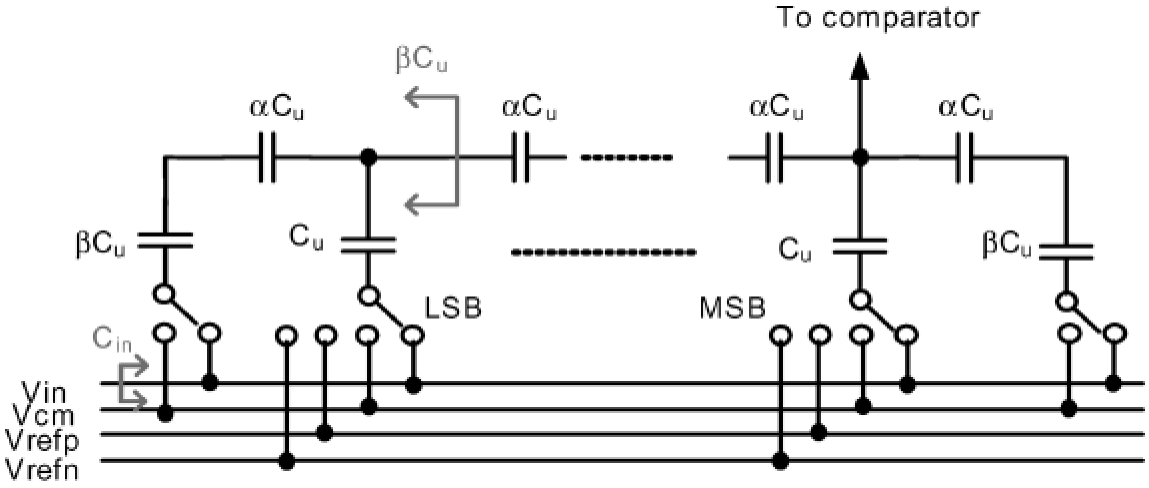
[W. Liu, P. Huang, and Y. Chiu, "A 12-bit, 45-MS/s, 3-mW redundant Successive-Approximation-Register Analog-to-Digital converter with digital calibration," IEEE J. Solid-State Circuits, vol. 46, no. 11, pp. 2661– 2672, Nov. 2011]

# EE 240C
# Analog-Digital Interface Integrated Circuits

# SAR ADC Redundancy

# Redundant Non-Binary DAC



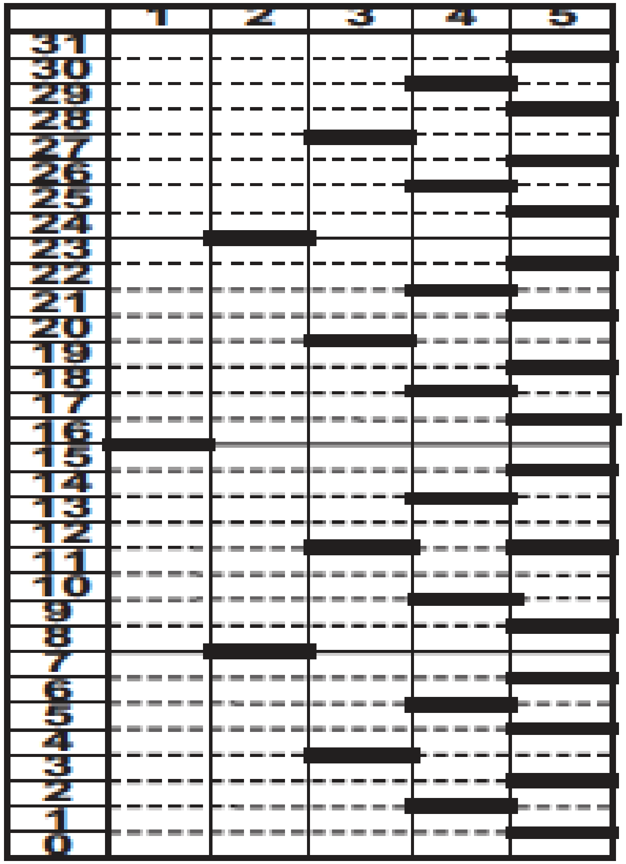


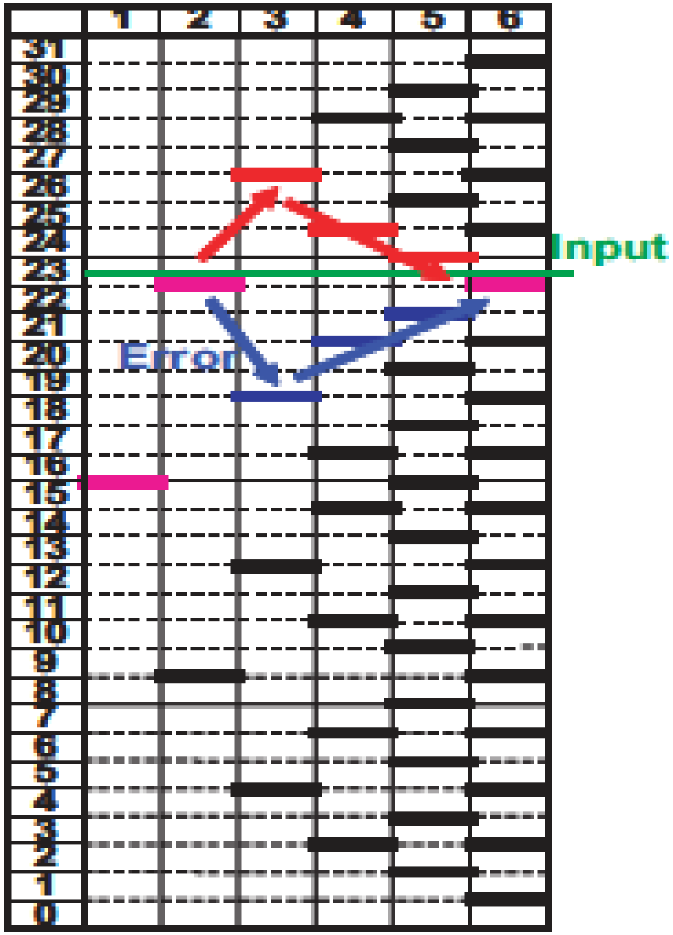
$$\begin{cases} \beta = 1 + \alpha \| \beta \\ \mathrm{Radix} = 1 + (\beta/\alpha) \end{cases}$$

- Non-binary DAC (with 1 < Radix < 2) allows for recovery of erroneous comparator decisions, due to:
  - Incomplete DAC settling
  - Comparator noise

# Non-binary Search

[Ogowa, “SAR ADC Algorithm with Redundancy”, 2008]

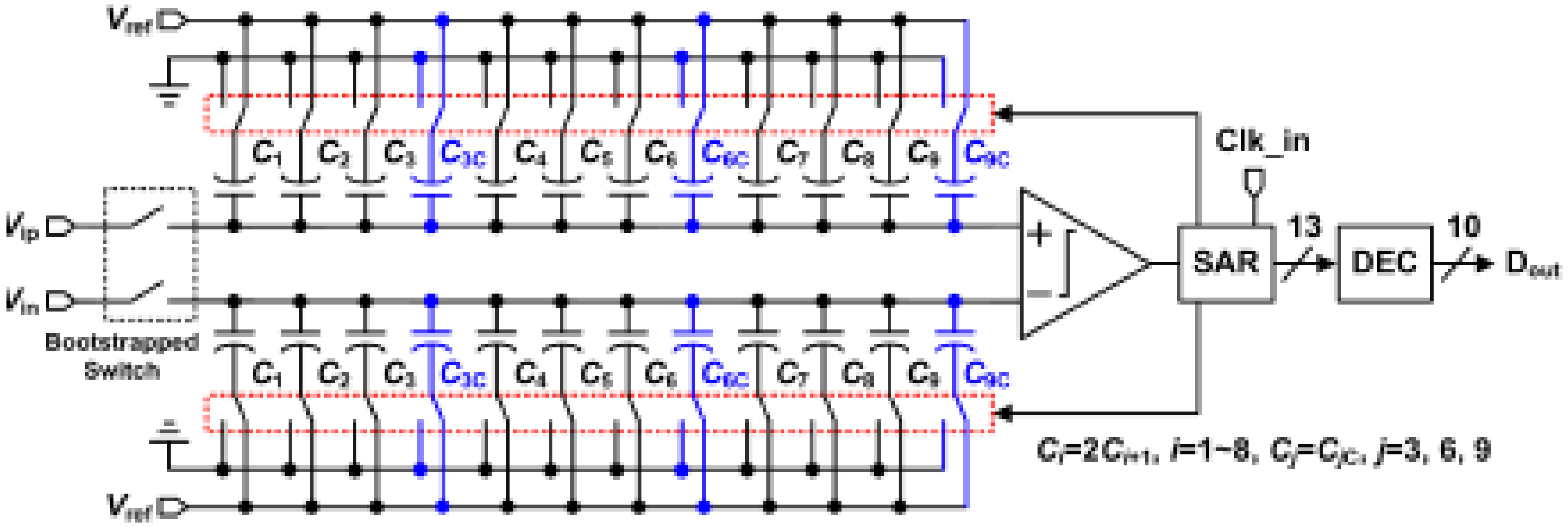




Binary Search (5b / 5 steps)

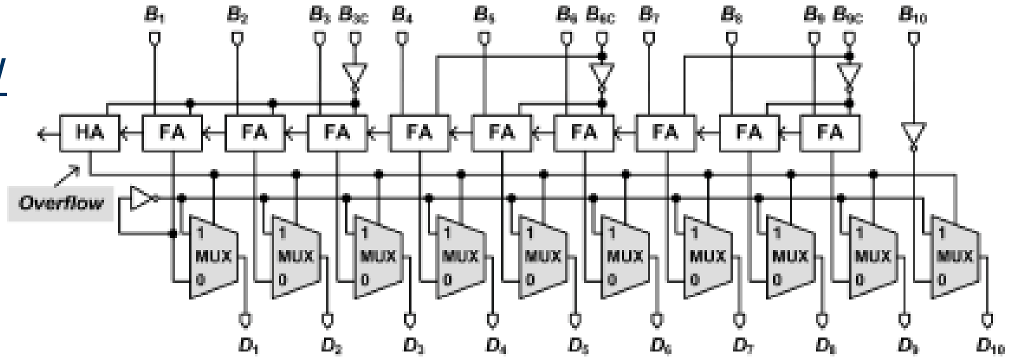
Non-Binary Search (5b / 6 steps)

# Redundant Binary DAC



$C_i=2C_{i+1}$, $i=1\sim8$, $C_j=C_{jC}$, $j=3, 6, 9$

[C.-C. Liu et al., “A 10b 100ms/s 1.13mW SAR ADC with binary-scaled error compensation,” in 2010 IEEE ISSCC, pp. 386–387]

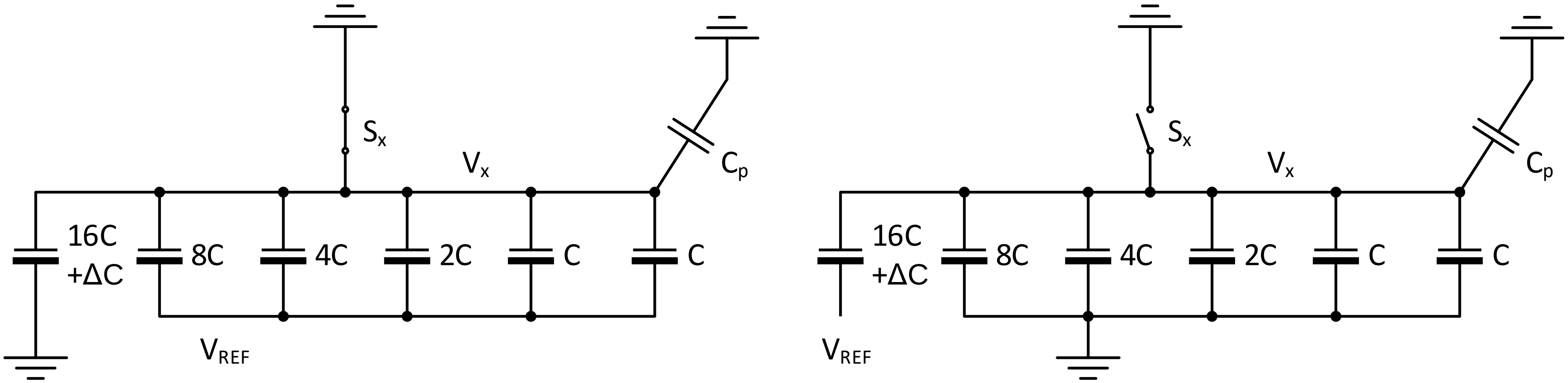


# EE 240C
# Analog-Digital Interface Integrated Circuits

# SAR ADC DAC Calibration

Johan Vanderhaegen
University of California, Berkeley
jpv@eecs.berkeley.edu

# DAC Calibration Scheme (5-bit Example)



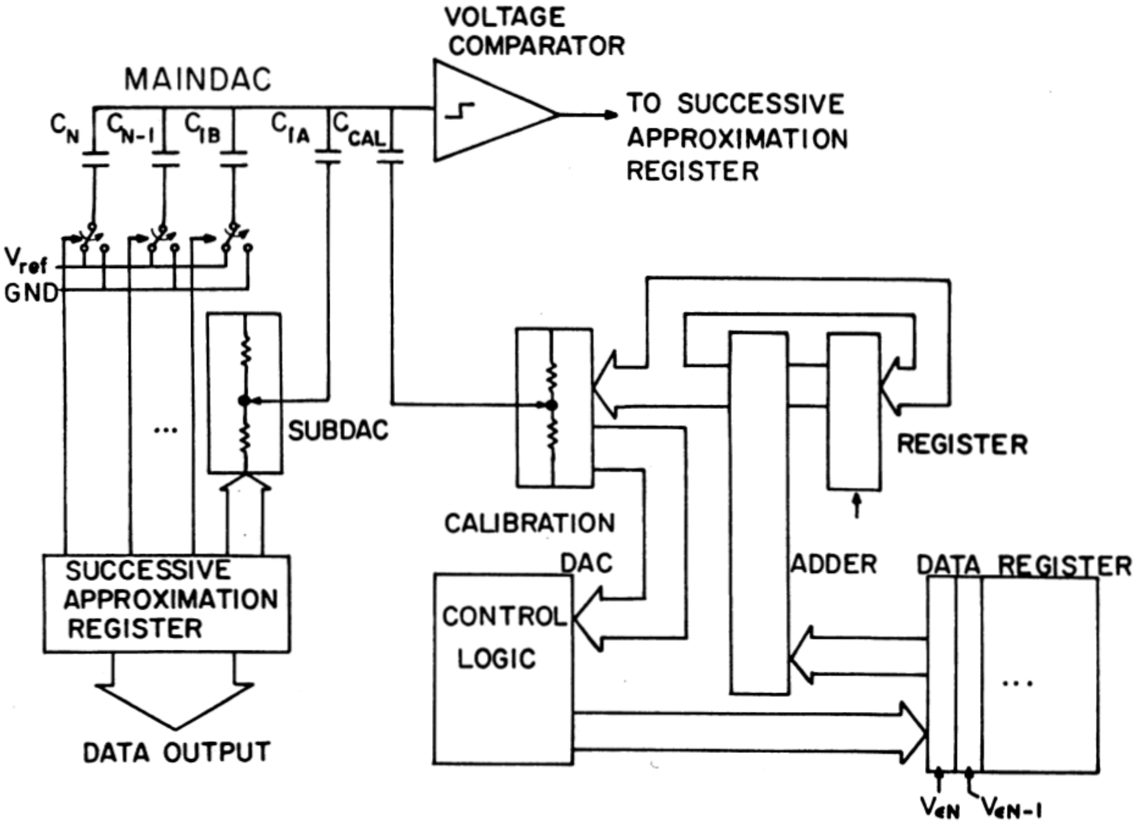
- Voltage $V_x$ after opening $S_x$ and switching cap voltages for MSB:

$$V_{x_1} = \frac{\Delta C}{C_{TOTAL}} V_{REF}$$

- Use calibration DAC to digitize residual and to correct conversion

[H.-S. Lee, “Self-calibration techniques for successive approximation analog- to-Digital converters, ” PhD thesis, University of California at Berkeley, 1984]

# DAC Calibration Scheme



[H.-S. Lee, “Self-calibration techniques for successive approximation analog- to-Digital converters, ” PhD thesis, University of California at Berkeley, 1984]

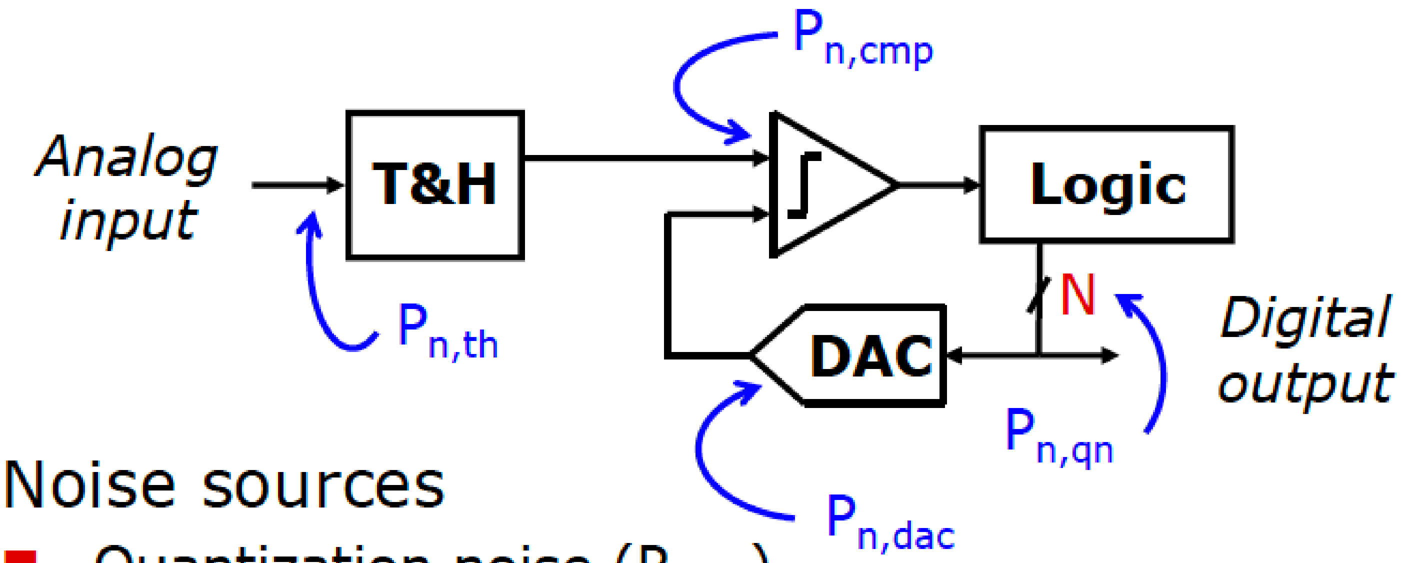
- Digital overhead

# EE 240C

# Analog-Digital Interface Integrated Circuits

# Successive Approximation ADC

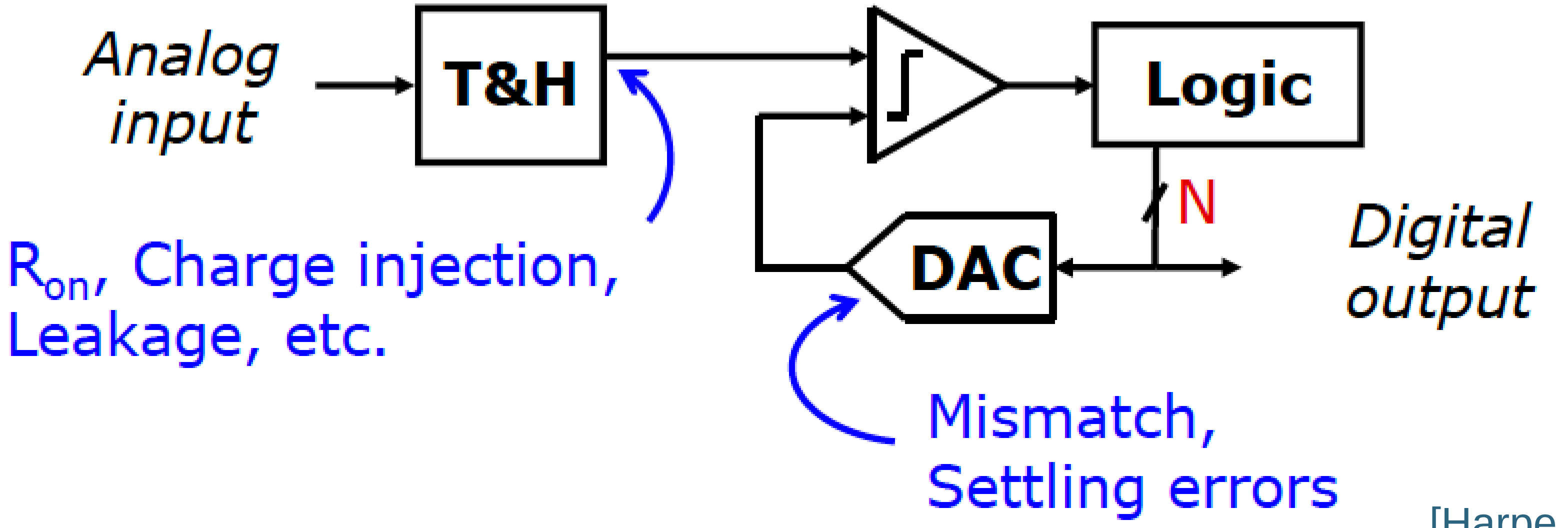
# SAR ADC Overall Noise



- Noise sources
  - Quantization noise ($P_{n,qn}$)
  - T&H ($P_{n,th}$)
  - DAC noise ($P_{n,dac}$)
  - Comparator noise ($P_{n,cmp}$)
- Total input-referred noise:
  - $P_{n,tot} \approx P_{n,qn} + P_{n,th} + P_{n,dac} + P_{n,cmp}$

[Harpe2016]

# SAR ADC Overall Linearity


Analog input
T&H
Logic
N
DAC
Digital output
$R_{on}$, Charge injection, Leakage, etc.
Mismatch, Settling errors

[Harpe2016]

- T&H, DAC, comparator offset → ADC offset
- T&H, DAC gain error → ADC gain error
- T&H, DAC non-linearity → ADC non-linearity

# SAR ADC Overall Speed

- $T_{clk} > T_{DAC} + T_{comp} + {}_{Tlogic}$

  $T_{DAC}$: DAC settling time
  $T_{comp}$: comparator delay
  $T_{logic}$: logic delay

- T&H bandwidth

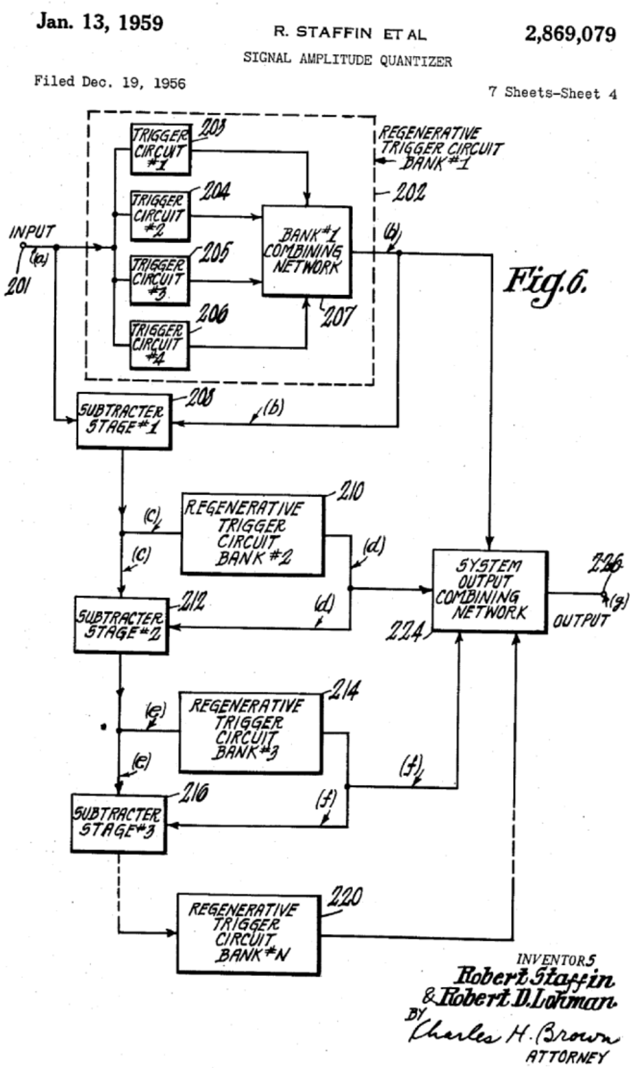
# EE 240C
# Analog-Digital Interface Integrated Circuits

# Pipeline ADC

# Outline

- Background
  - History and state-of the art performance
  - General idea of multi-step A/D conversion
- Pipeline ADC basics
  - Ideal block diagram and operation, impact of block nonidealities
- Ways to deal with nonidealities
  - Redundancy, calibration
- CMOS implementation details
  - Stage scaling, MDAC design
- Architectural options
  - OTA sharing, SHA-less front-end
- Research topics

# History (1)



Jan. 13, 1959 R. STAFFIN ET AL 2,869,079

SIGNAL AMPLITUDE QUANTIZER

Filed Dec. 19, 1956 7 Sheets-Sheet 4

INVENTORS
Robert Staffin
& Robert D. Lohman
BY
Charles H. Brown
ATTORNEY

US Patent # 2,869,079: Staffin and Lohman, "Signal Amplitude Quantizer," 1959

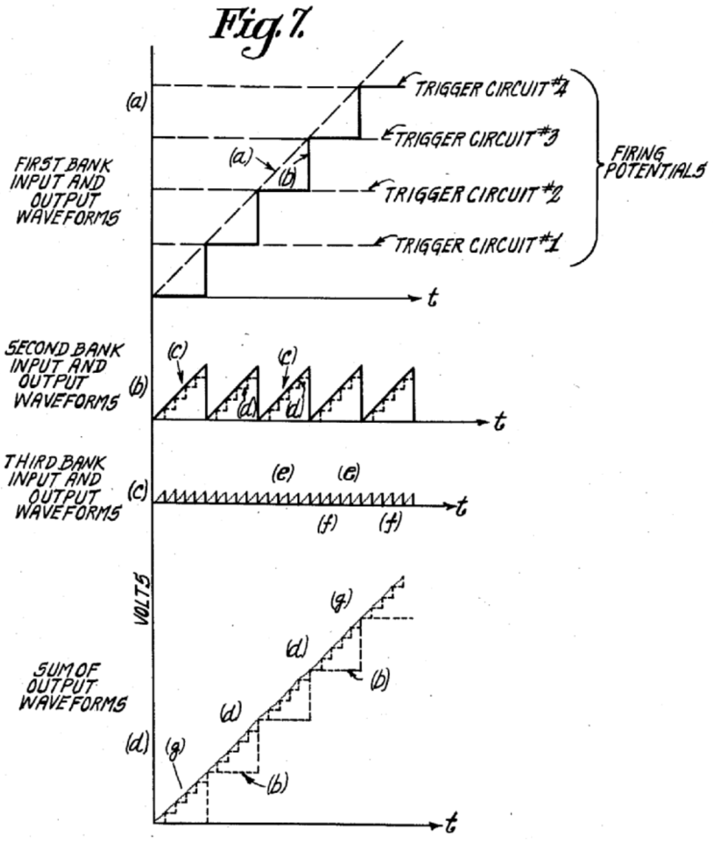


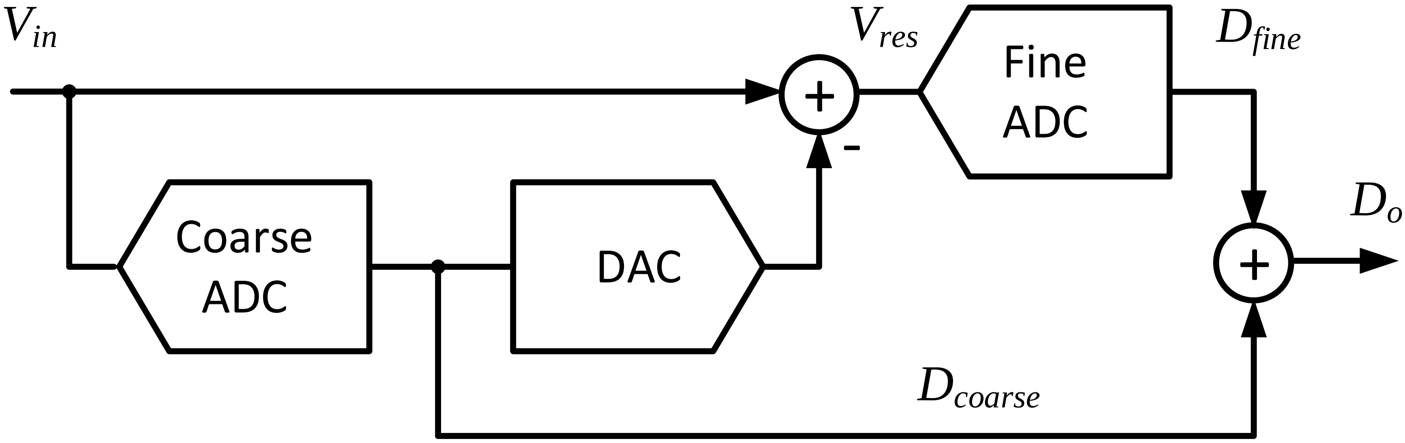
Fig. 7.

# History (2)

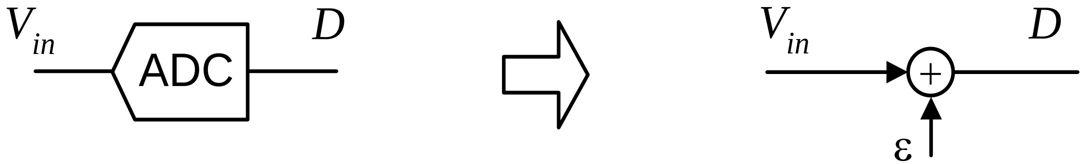
- First multi-step ADC with "error correction"
  - T.C. Verster, "A method to increase the accuracy of fast Serial-Parallel Analog-to-Digital Converters," IEEE Trans. Electronic Computers, EC-13, pp. 471-473, 1964.
- First pipeline ADC
  - B.D. Smith, "An Unusual Electronic Analog-Digital Conversion Method," IRE Transactions on Instrumentation, pp.155-160, June 1956.
- First pipeline ADCs in CMOS
  - S.H. Lewis and P.R. Gray, "A pipelined 5-Msample/s 9-bit analog-to-digital converter," JSSC, pp. 954-961, Dec. 1987.
  - S. Sutardja and P.R. Gray, "A pipelined 13-bit 250-ks/s 5-V analog-to-digital converter," JSSC, pp. 1316-1323, Dec. 1988.

# General Concept of Multi-Step Conversion

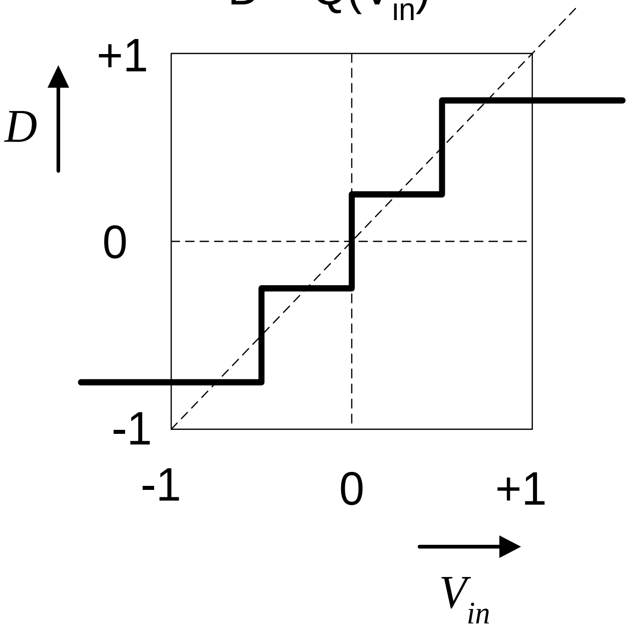
- General idea (two-step example)
  1. Perform a "coarse" quantization of the input
  2. Compute residuum (error) of step 1 conversion using a DAC and subtractor
  3. Digitize computed residuum using a second "fine" quantizer and digitally add to output



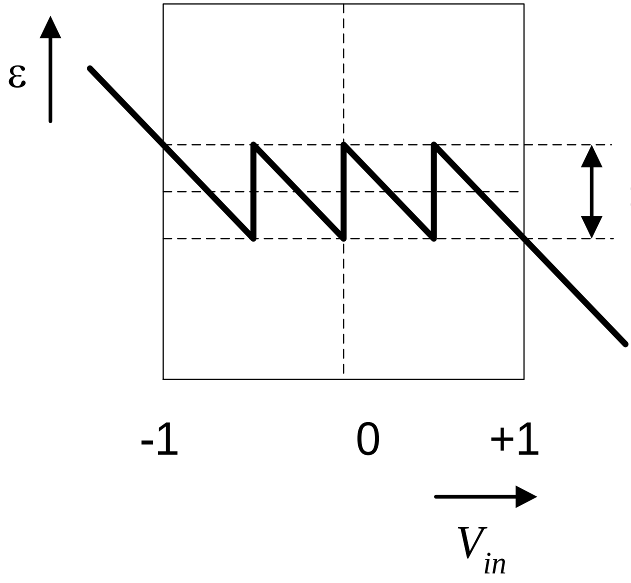
# Quantizer Model



Transfer function
D = Q($V_{in}$)



Quantization Error
ε= D-$V_{in}$

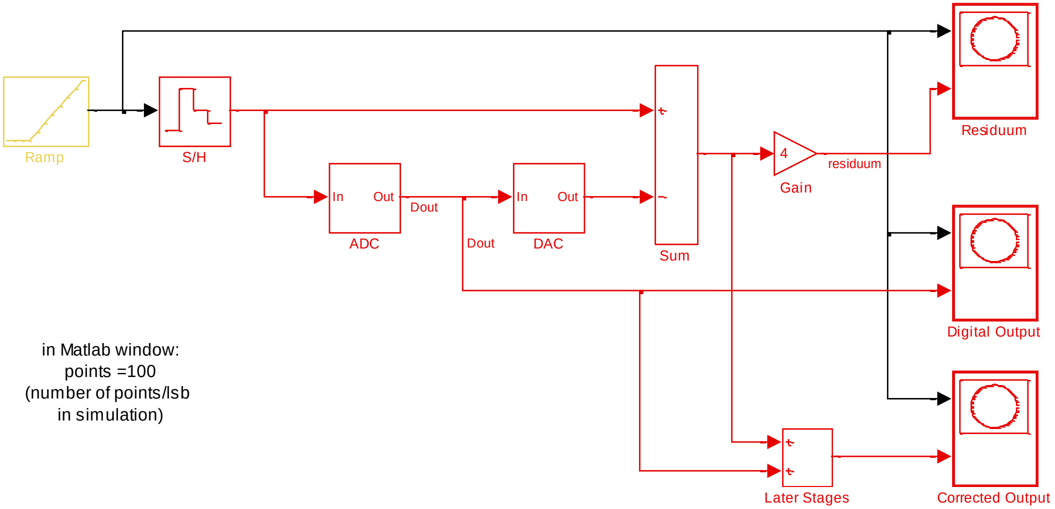


$$\pm\frac{1}{2}LSB = \pm\frac{1}{2}\frac{2}{2^B}$$

# EE 240C
# Analog-Digital Interface Integrated Circuits

# Pipeline Simulation

# Pipeline Simulation Model

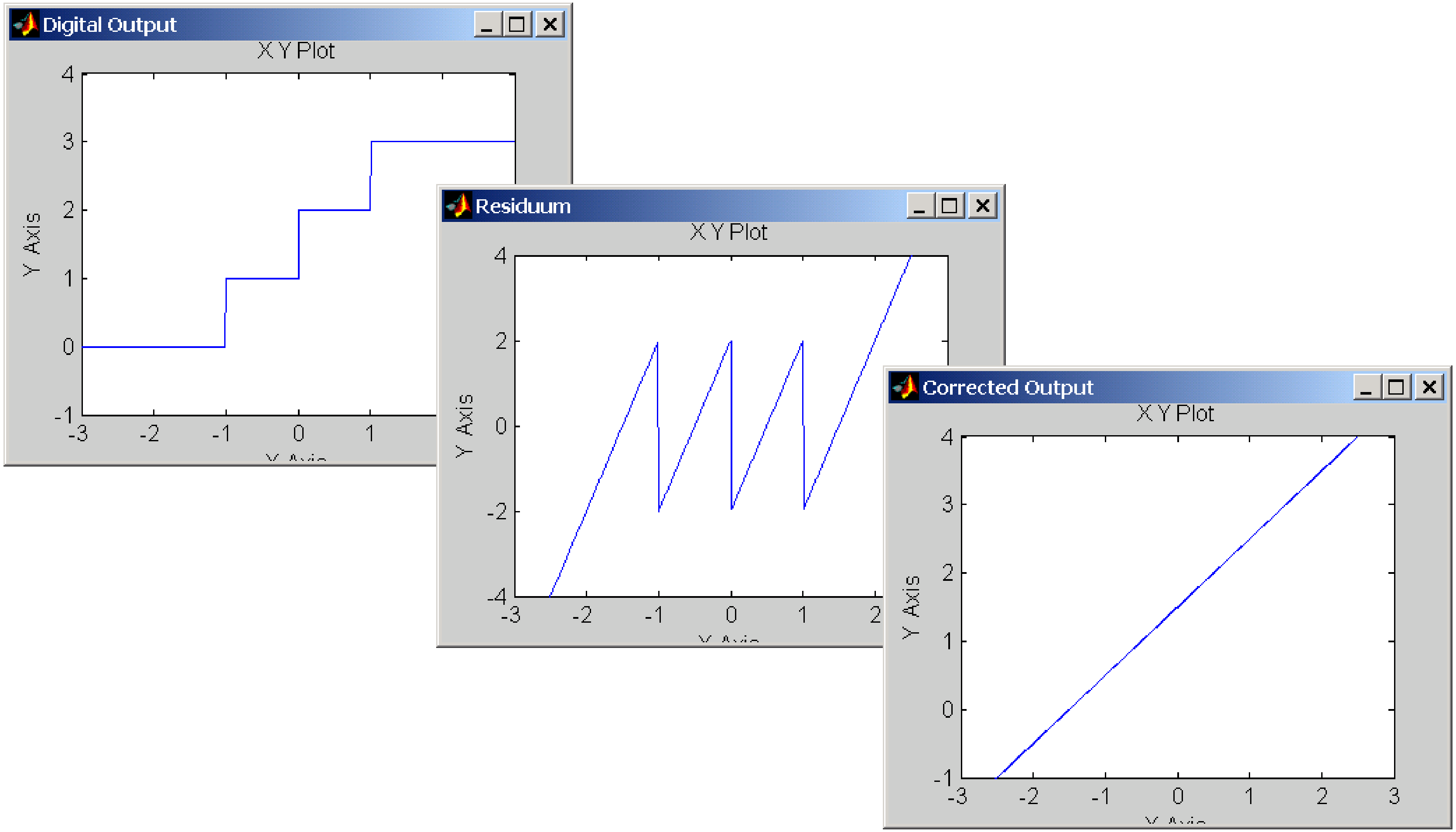


in Matlab window:
points =100
(number of points/lsb
in simulation)

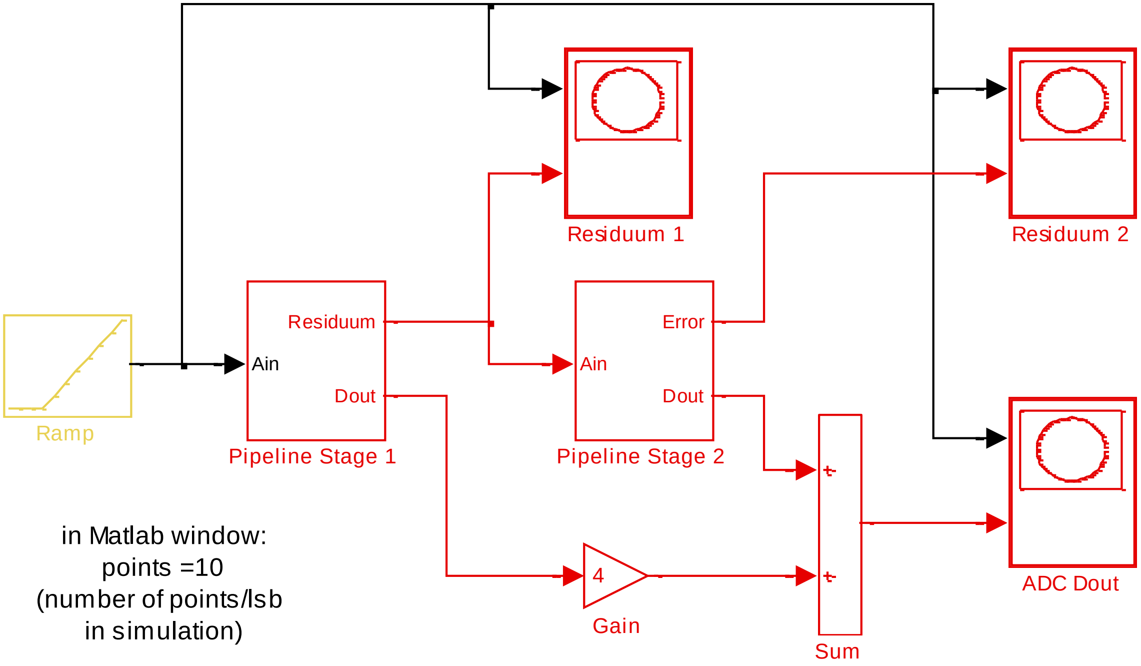
$$V_{\mathrm{res}} = G(V_{\mathrm{in}} - D\Delta)$$

See Matlab/Simulink L18_pipe_3_el.mdl

# Simulation of 2-Bit Stage



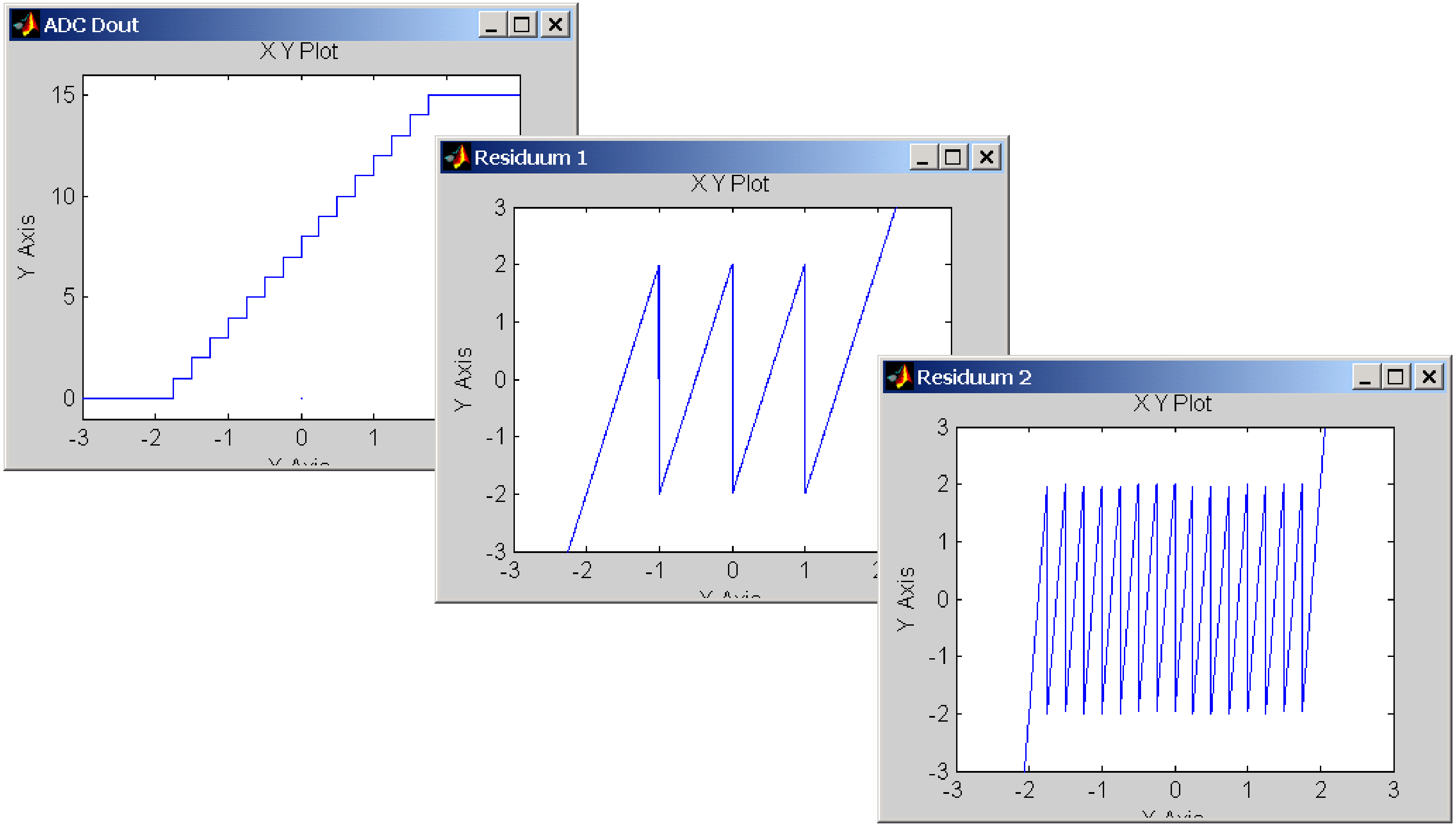
# Pipeline ADC Model



in Matlab window:
points =10
(number of points/lsb
in simulation)

See Matlab/Simulink L18_pipe_2bps_error.mdl

# Simulation Result

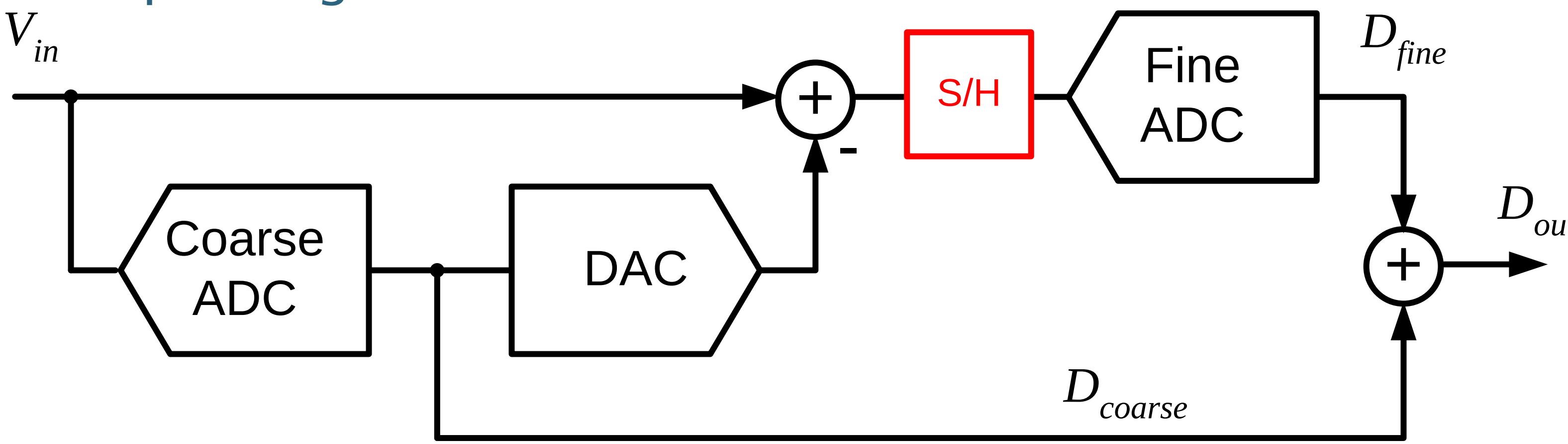


# EE 240C
# Analog-Digital Interface Integrated Circuits

## Pipelining

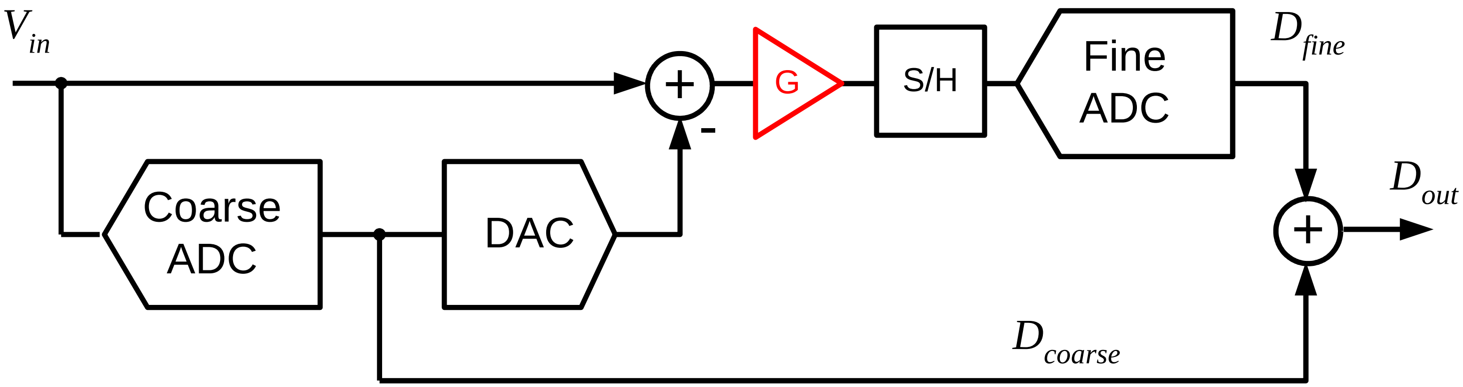
# S/H

- Conversion time is proportional to number of stages employed
  - E.g. for a two-step ADC, time required for conversion is $T_{conv} = 2 \cdot T_{A/D} + T_{D/A} + T_{SUB}$
- Solution
  - Introduce a sample and hold operation after subtraction
  - Fine ADC has one full clock cycle until new residuum becomes available
  - "Pipelining"

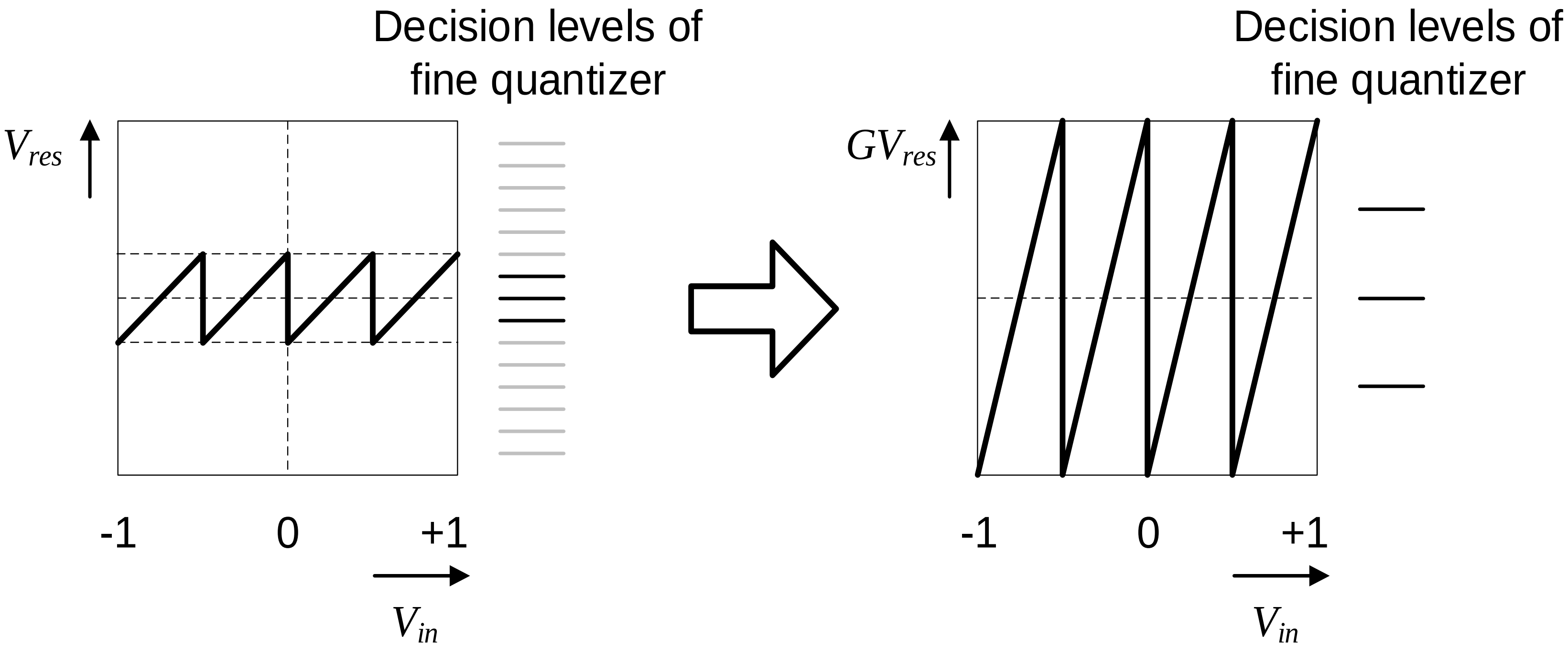


# Interstage Gain

- Fine ADC(s) must have precision commensurate with overall target resolution
  - E.g. 8-bit converter with 4-bit/4-bit partition; fine 4-bit decision levels must have "8-bit precision"
- Solution
  - Introduce gain after subtraction



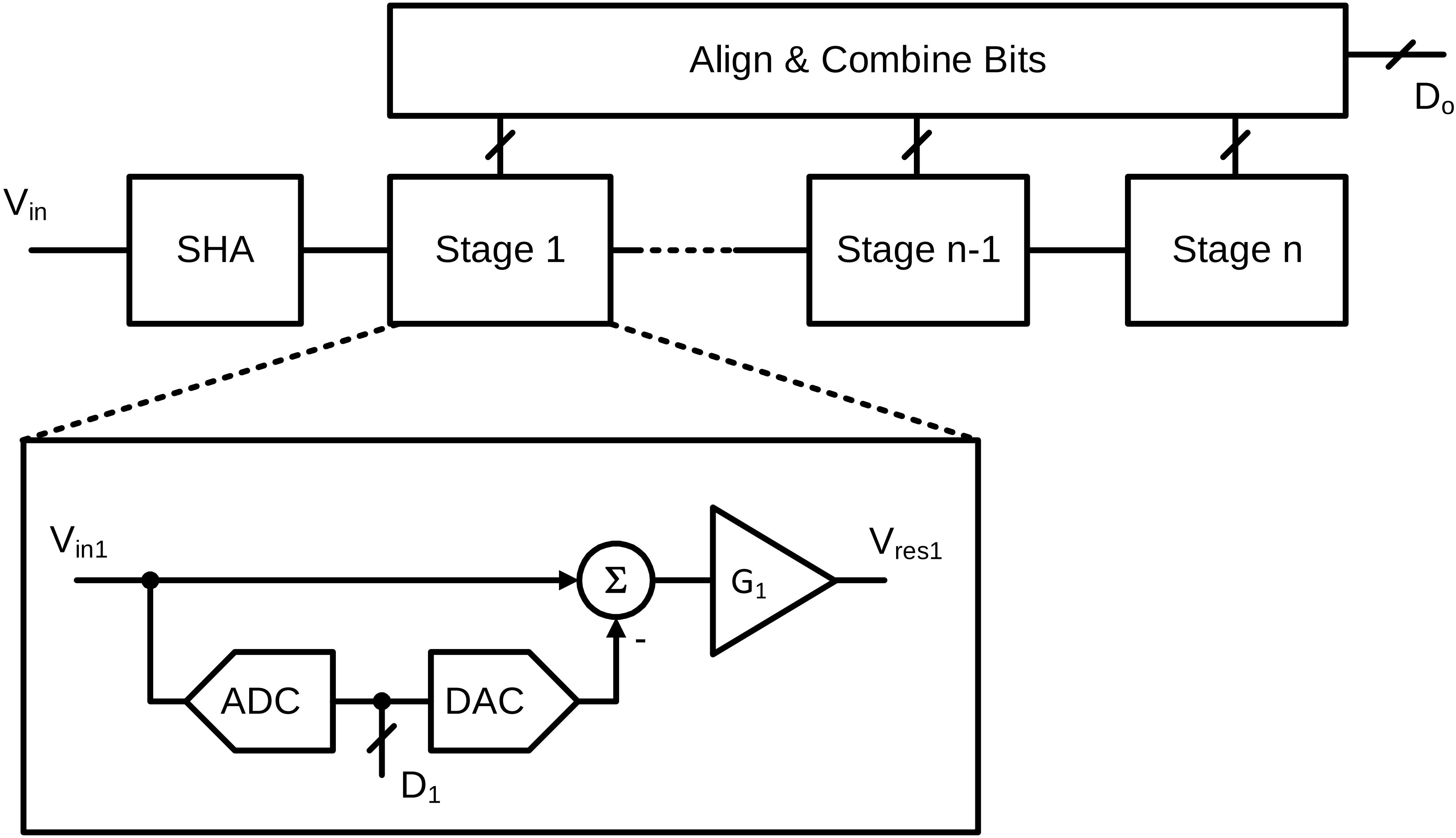
# Input to Fine Quantizer with Gain


Decision levels of fine quantizer

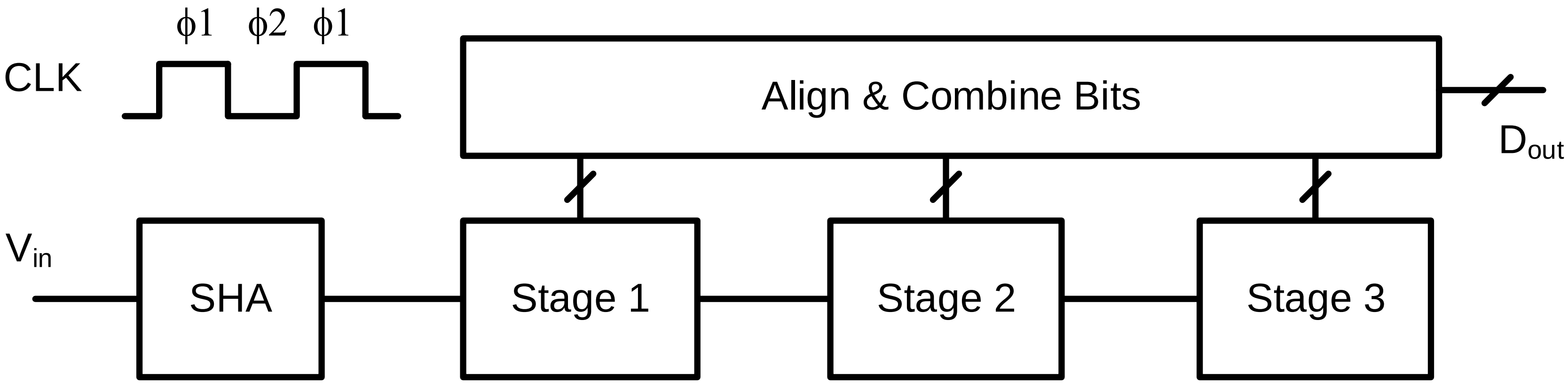
Decision levels of fine quantizer

- No longer need precision comparators

# Pipeline ADC Block Diagram



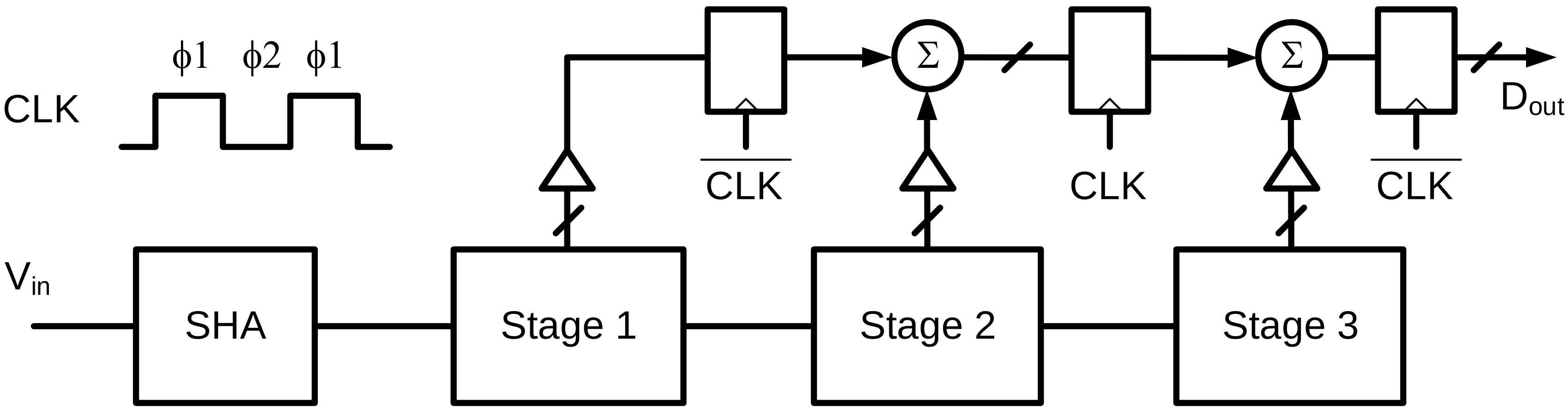
# Concurrent Stage Operation



| | | | | |
|---|---|---|---|---|
| ϕ1 | ACQUIRE | CONVERT | ACQUIRE | CONVERT |
| ϕ2 | BUFFER | ACQUIRE | CONVERT | ACQUIRE |

- Stages operate on the input signal like a shift register
- New output data every clock cycle, but each stage introduces ½ clock cycle latency

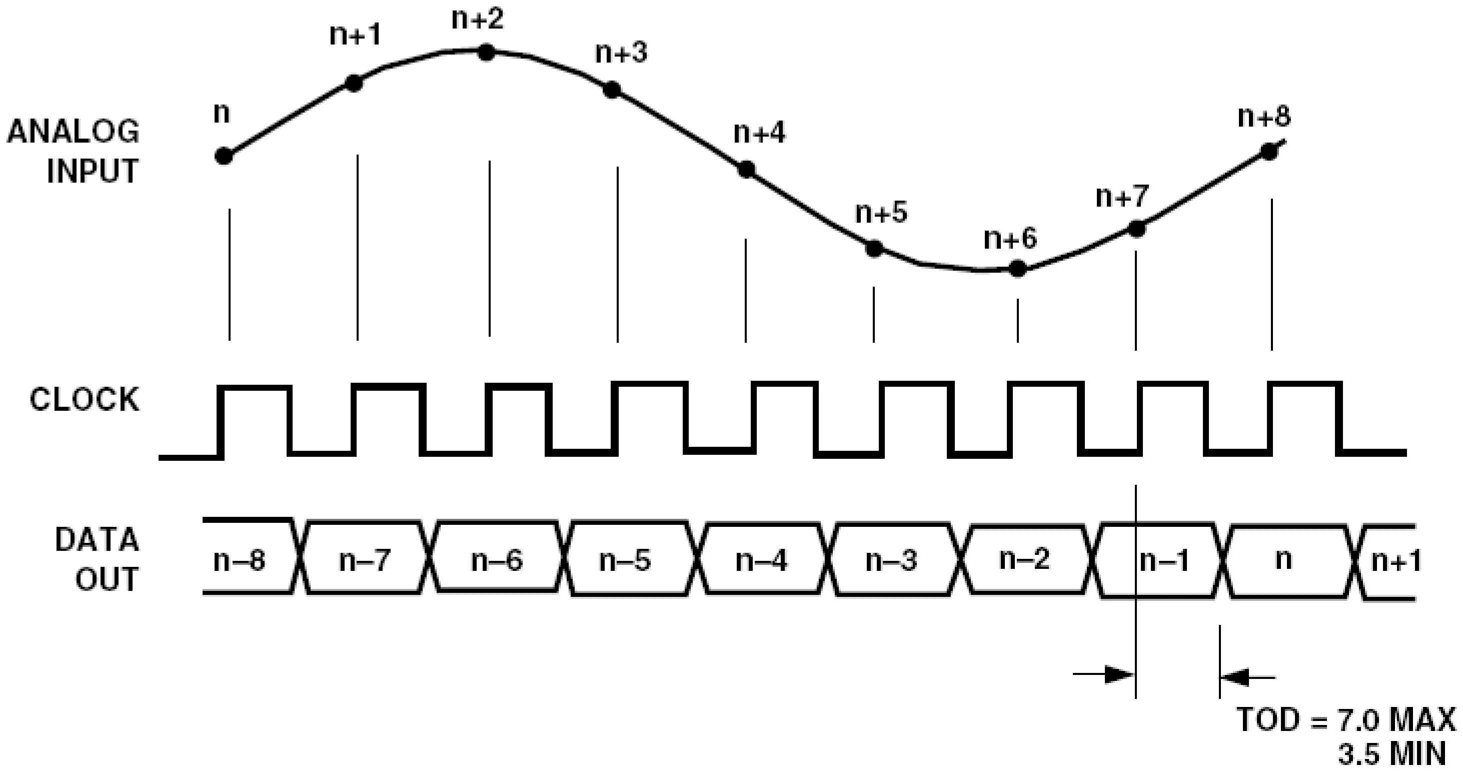
# Data Alignment



| | | | | |
|---|---|---|---|---|
| ϕ1 | ACQUIRE | CONVERT | ACQUIRE | CONVERT |
| ϕ2 | BUFFER | ACQUIRE | CONVERT | ACQUIRE |

- Digital shift register aligns sub-conversion results in time
- Digital output is taken as weighted sum of stage bits

# Latency



[Analog Devices, AD9226 Data Sheet]

# Pipeline ADC Characteristics

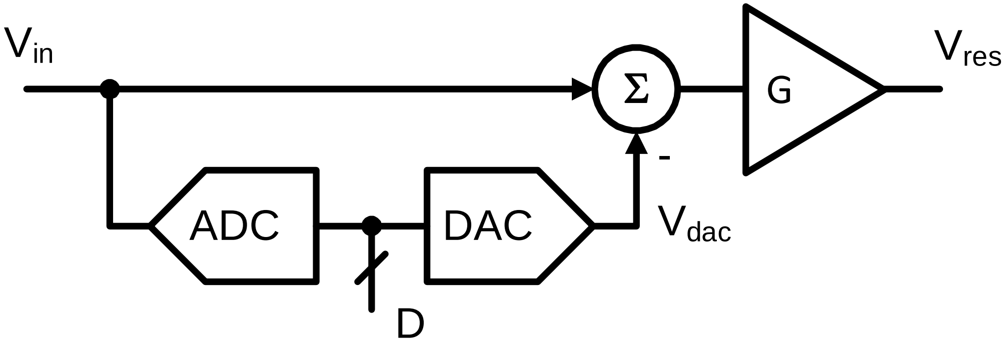
- Number of components grows linearly with resolution
  - Unlike flash ADC, where components ~ $2^B$
- Pipeline ADC trades latency for conversion speed
  - Throughput limited by speed of one stage
    - Enables high-speed operation
  - Latency can be an issue in some applications
    - E.g. in feedback control loops
- Pipelining requires good analog "memory elements"
  - Calls for implementation in CMOS using switched-capacitor circuits

# EE 240C
# Analog-Digital Interface Integrated Circuits

# Pipeline Modeling

# Stage Analysis

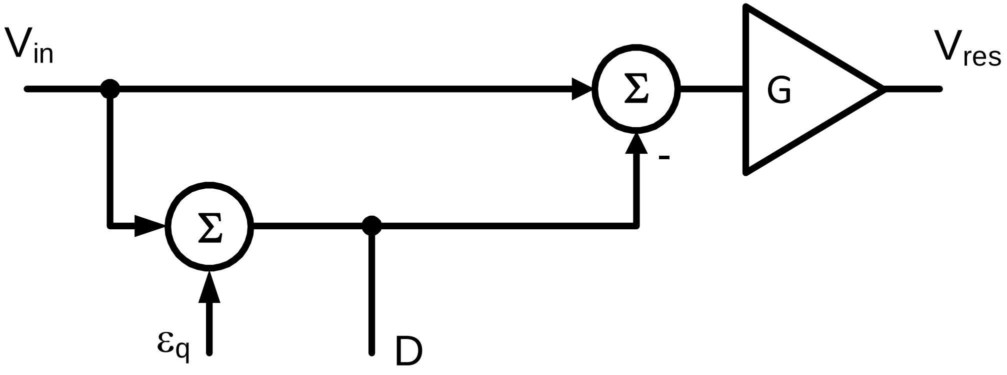
- Ignore timing/clock delays for simplicity



$$D = Q(V_{\text{in}})$$

$$V_{res} = G \cdot [V_{\text{in}} - V_{\text{dac}}]$$

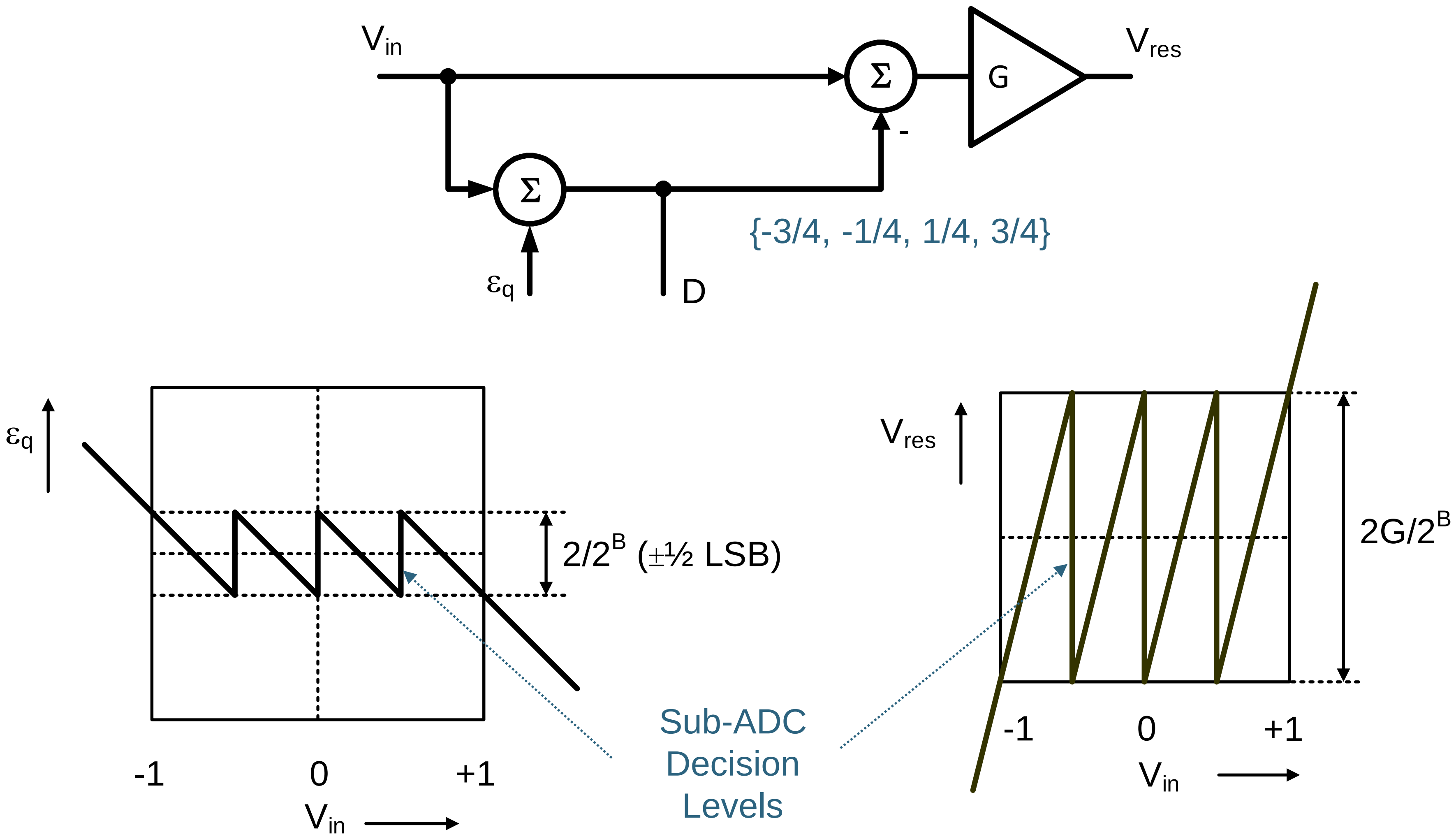
# Stage Model with Ideal DAC



$$D = V_{\text{in}} + \varepsilon_q \qquad V_{\text{res}} = -G \cdot \varepsilon_q$$

- Residue of pipeline stage ($V_{res}$) is equal to (–gain) times sub–ADC quantization error

# "Residue Plot" (2-bit Sub-ADC)

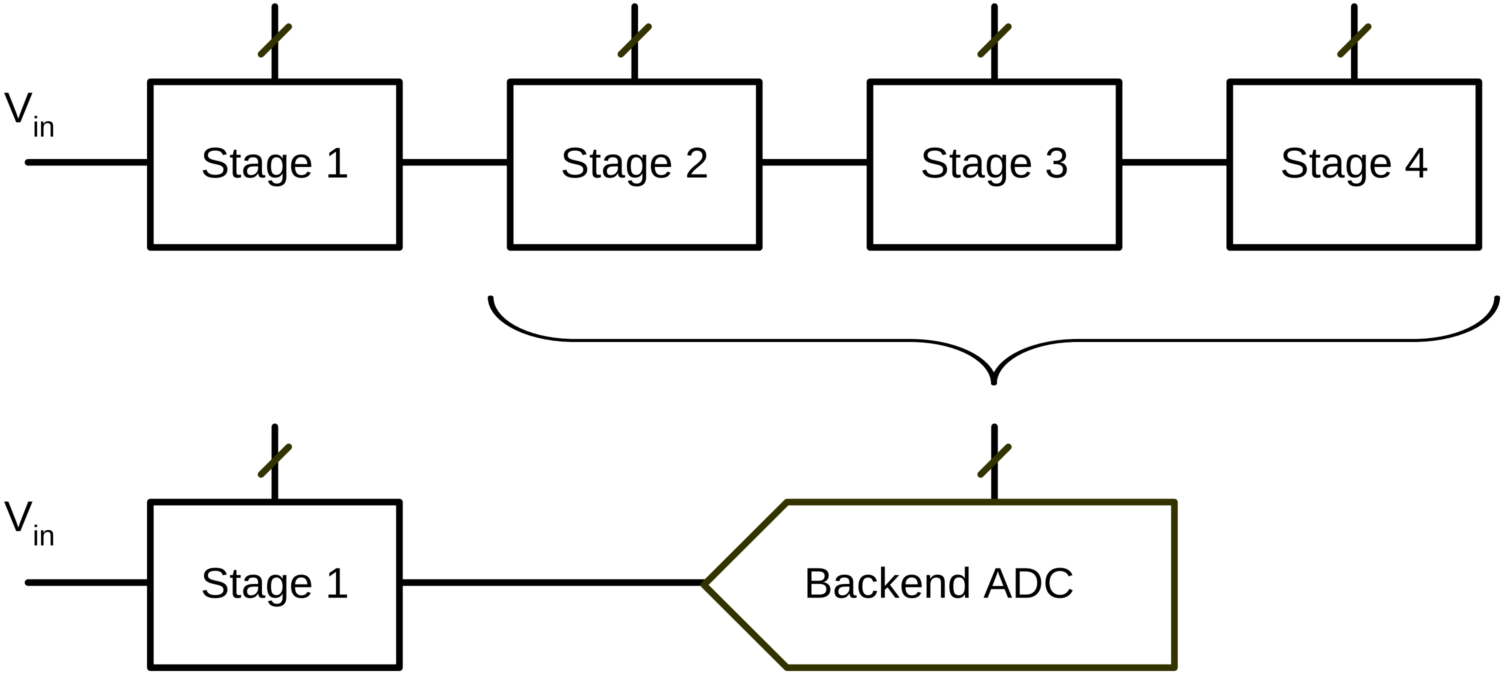


{-3/4, -1/4, 1/4, 3/4}

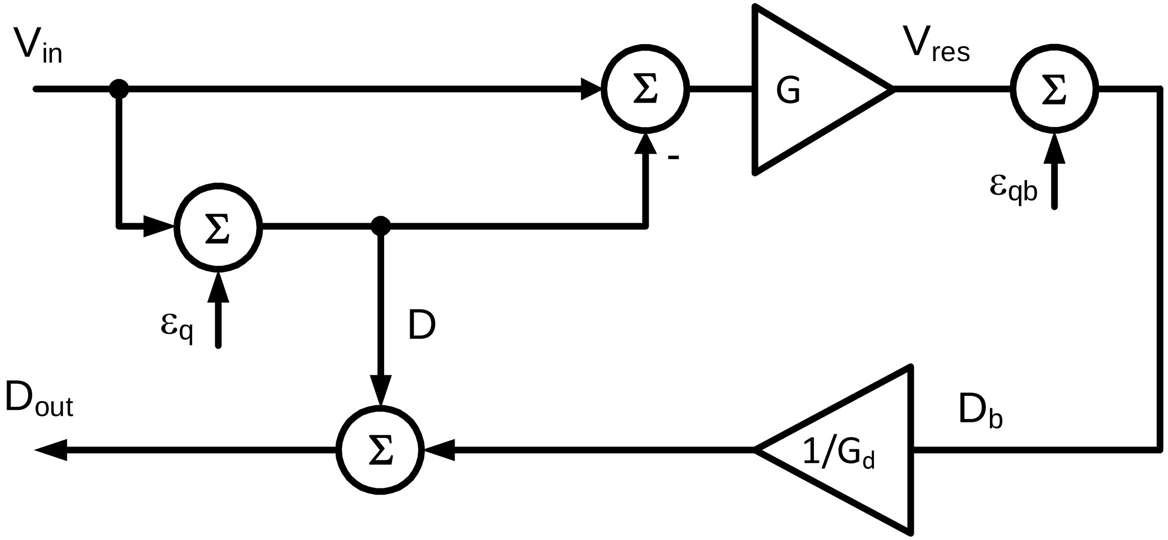
Sub-ADC Decision Levels

# Pipeline Decomposition

- Often convenient to look at pipeline as single stage plus backend ADC



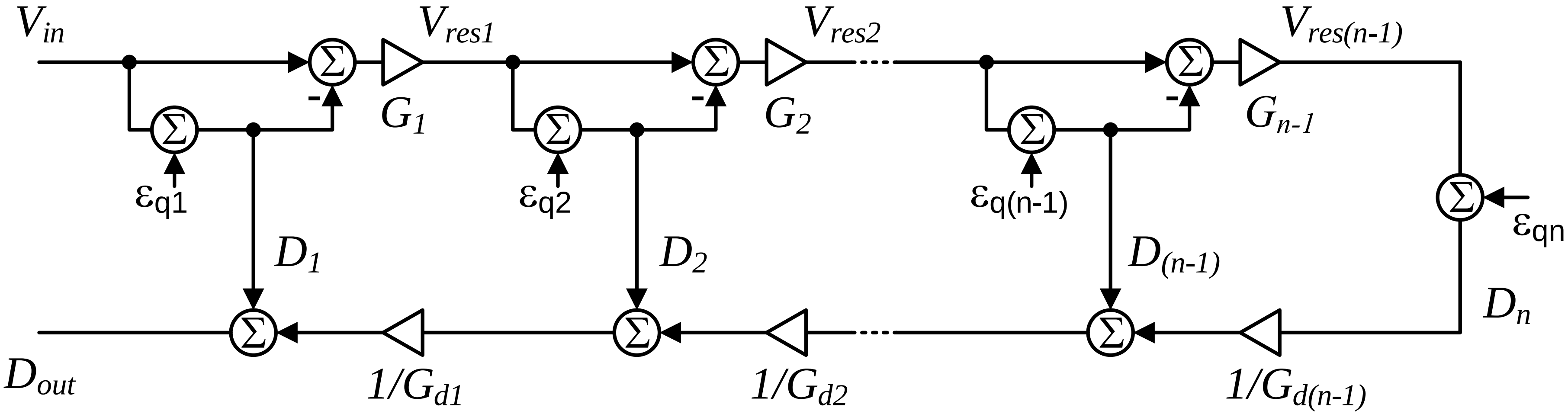
# Resulting Model



$$D_{\text{out}} = V_{\text{in}} + \varepsilon_q \left(1 - \cancel{\frac{G}{G_d}}\right) + \frac{\varepsilon_{qb}}{G_d}$$

With $G_d$=G

# Canonical Extension



$$D_{\text{out}} = V_{\text{in}} + \varepsilon_{q1}\left(1 - \frac{G_1}{G_{d1}}\right) + \frac{\varepsilon_{q2}}{G_{d1}}\left(1 - \frac{G_2}{G_{d2}}\right) + \ldots + \frac{\varepsilon_{q(n-1)}}{\prod_{j=1}^{n-2} G_{dj}}\left(1 - \frac{G_{(n-1)}}{G_{d(n-1)}}\right) + \frac{\varepsilon_{qn}}{\prod_{j=1}^{n-1} G_{dj}}$$

- First stage has most stringent precision requirements
- Note that above model assumes that all stages use same reference voltage (same full scale range)
  - This is true for most designs, one exception is [Limotyrakis 2005]

# General Result – Ideal Pipeline ADC

- With ideal DACs and ideal digital weights ($G_{dj}=G_j$)

$$D_{\text{out}} = V_{\text{in}} + \frac{\varepsilon_{qn}}{\prod_{j=1}^{n-1} G_j} \quad \Rightarrow \quad B_{\text{ADC}} = B_n + \sum_{j=1}^{n-1} \log_2 G_j$$

- The only error in $D_{\text{out}}$ is that of last quantizer, divided by aggregate gain
- Aggregate ADC resolution is independent of sub-ADC resolutions in stage 1…n-1 (!)
- Makes sense to define "effective" resolution of jth stage as $R_j=\log_2(G_j)$

# EE 240C

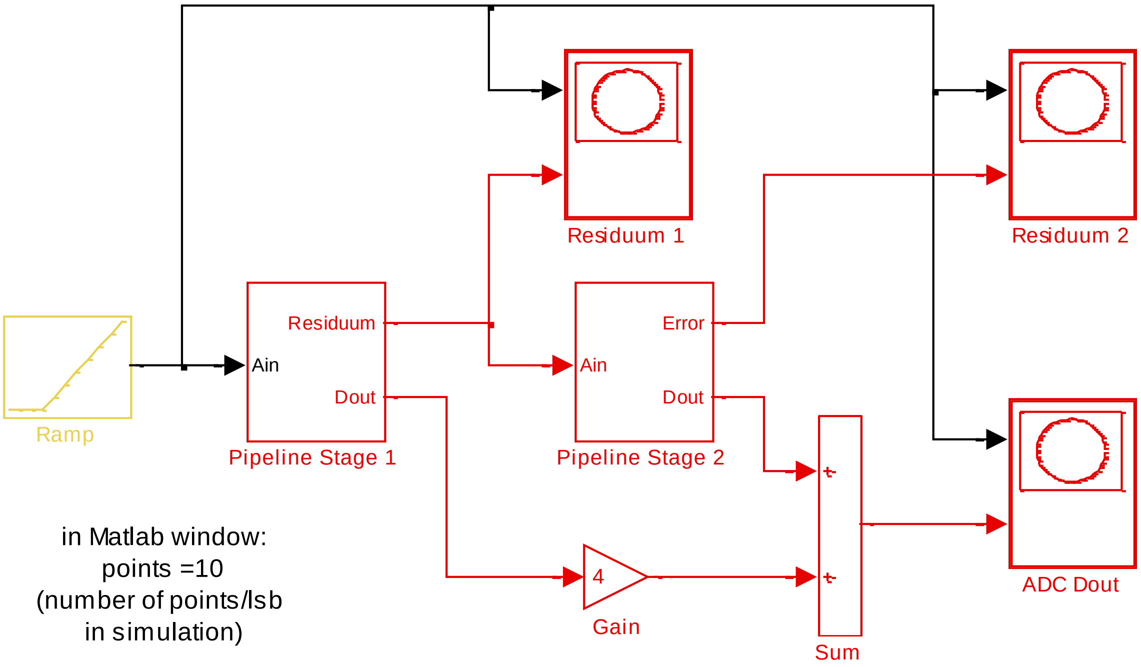
## Analog-Digital Interface Integrated Circuits

# Pipeline Design & Error Sources

# Questions

- How to pick stage gain G for a given sub-ADC resolution?
- Impact and compensation of nonidealities?
  - Sub-ADC errors
  - Amplifier offset
  - Amplifier gain error
  - Sub-DAC error
- Begin to explore these questions using a simple example
  - First stage with 2-bit sub-ADC, followed by 2-bit backend

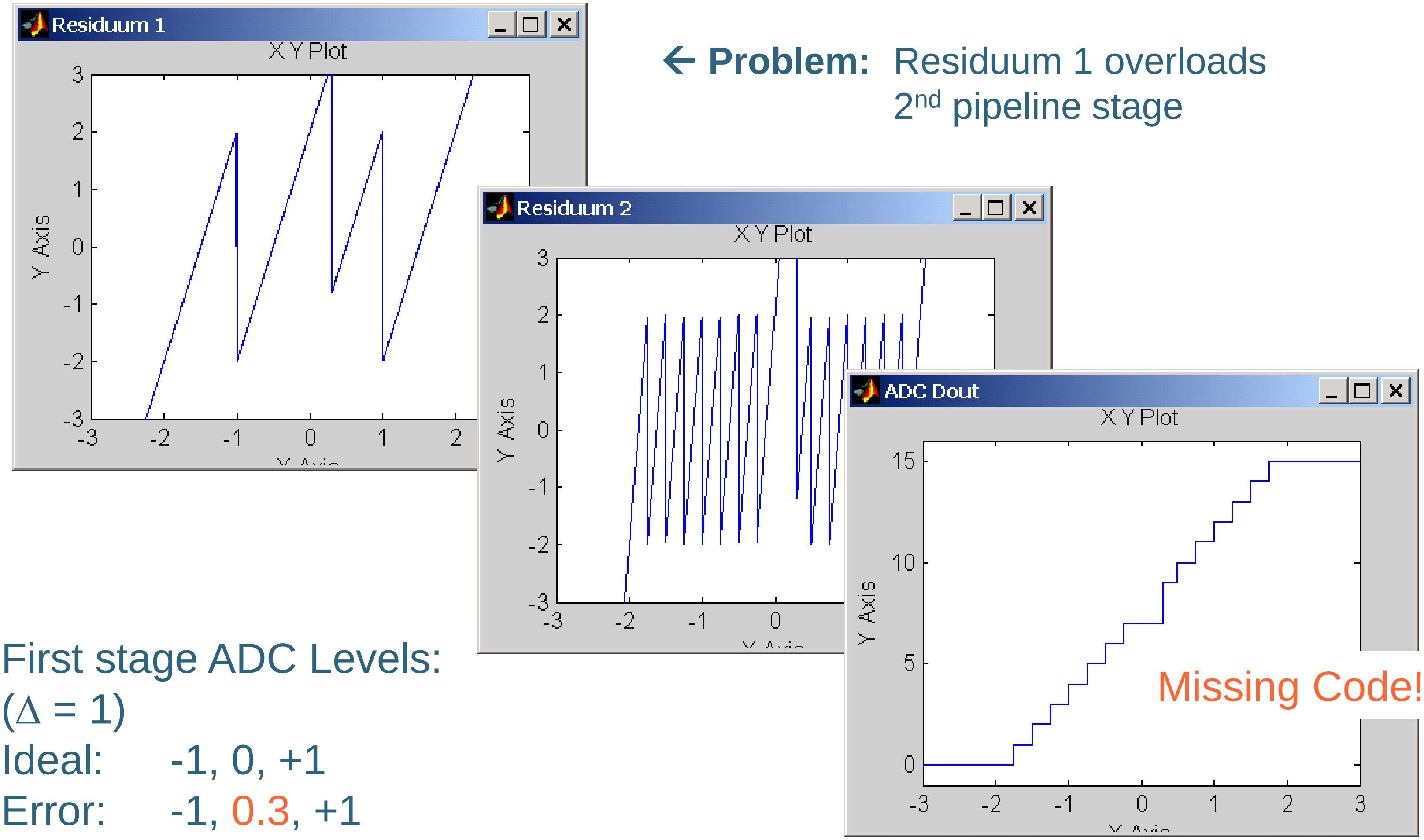
# Comparator Offset



in Matlab window:
points =10
(number of points/lsb
in simulation)

See Matlab/Simulink L18_pipe_2bps_error.mdl

# Comparator Offset



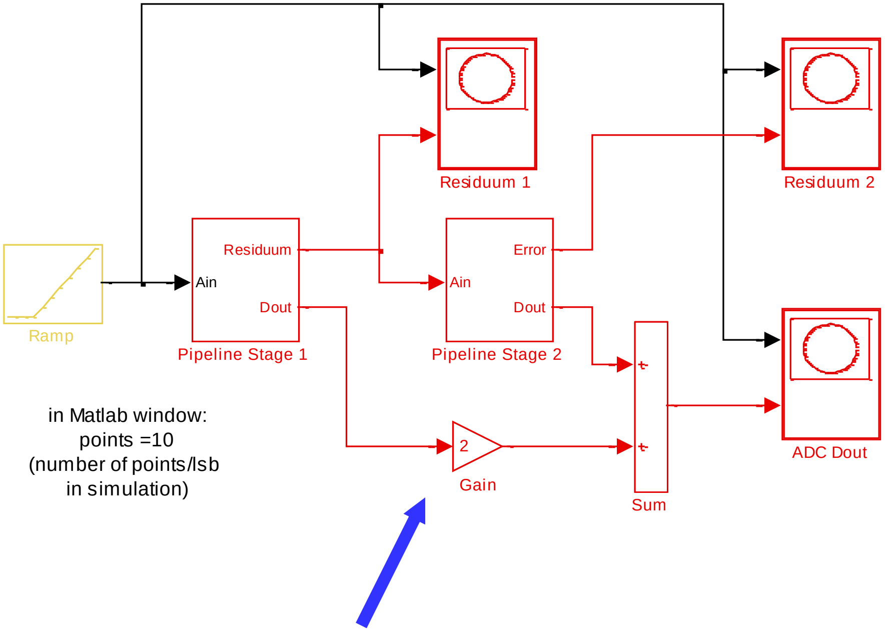
Missing Code!

← **Problem:** Residuum 1 overloads 2nd pipeline stage

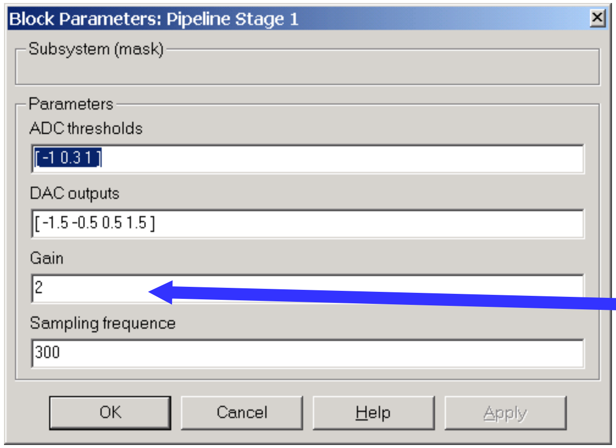
First stage ADC Levels:
(Δ = 1)
Ideal: -1, 0, +1
Error: -1, 0.3, +1

# Digital Correction

**Pipeline ADC, 2 bits per stage**
**Interstage gain = 2 for digital correction**



in Matlab window:
points =10
(number of points/lsb
in simulation)



Block Parameters: Pipeline Stage 1

Subsystem (mask)

Parameters

ADC thresholds

[-1 0.3 1]

DAC outputs

[-1.5 -0.5 0.5 1.5 ]

Gain

2

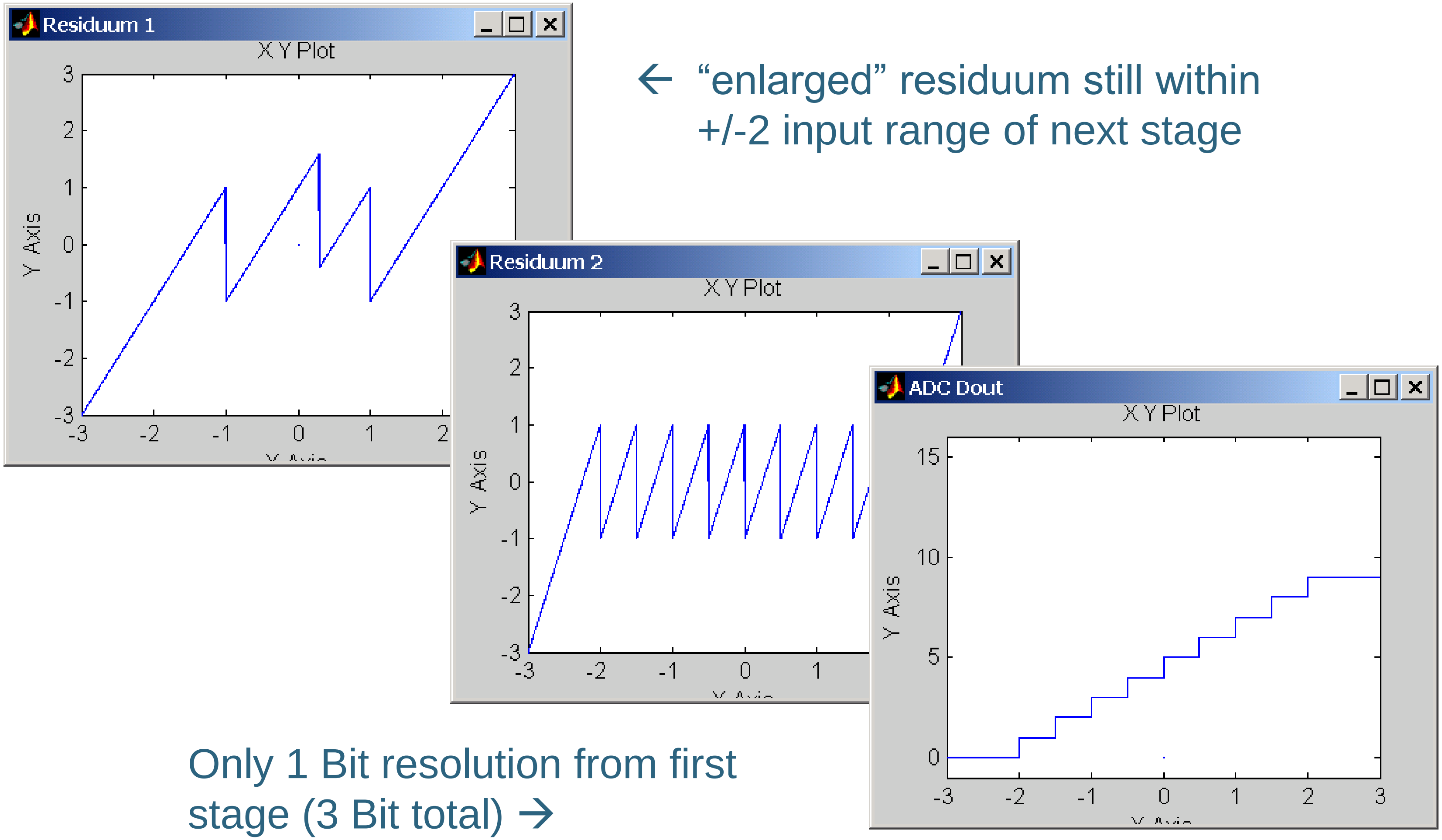
Sampling frequence

300

Reduced interstage gain:

- No overload (due to comparator offset)
- Reduced input (only 1 bit resolution per stage)

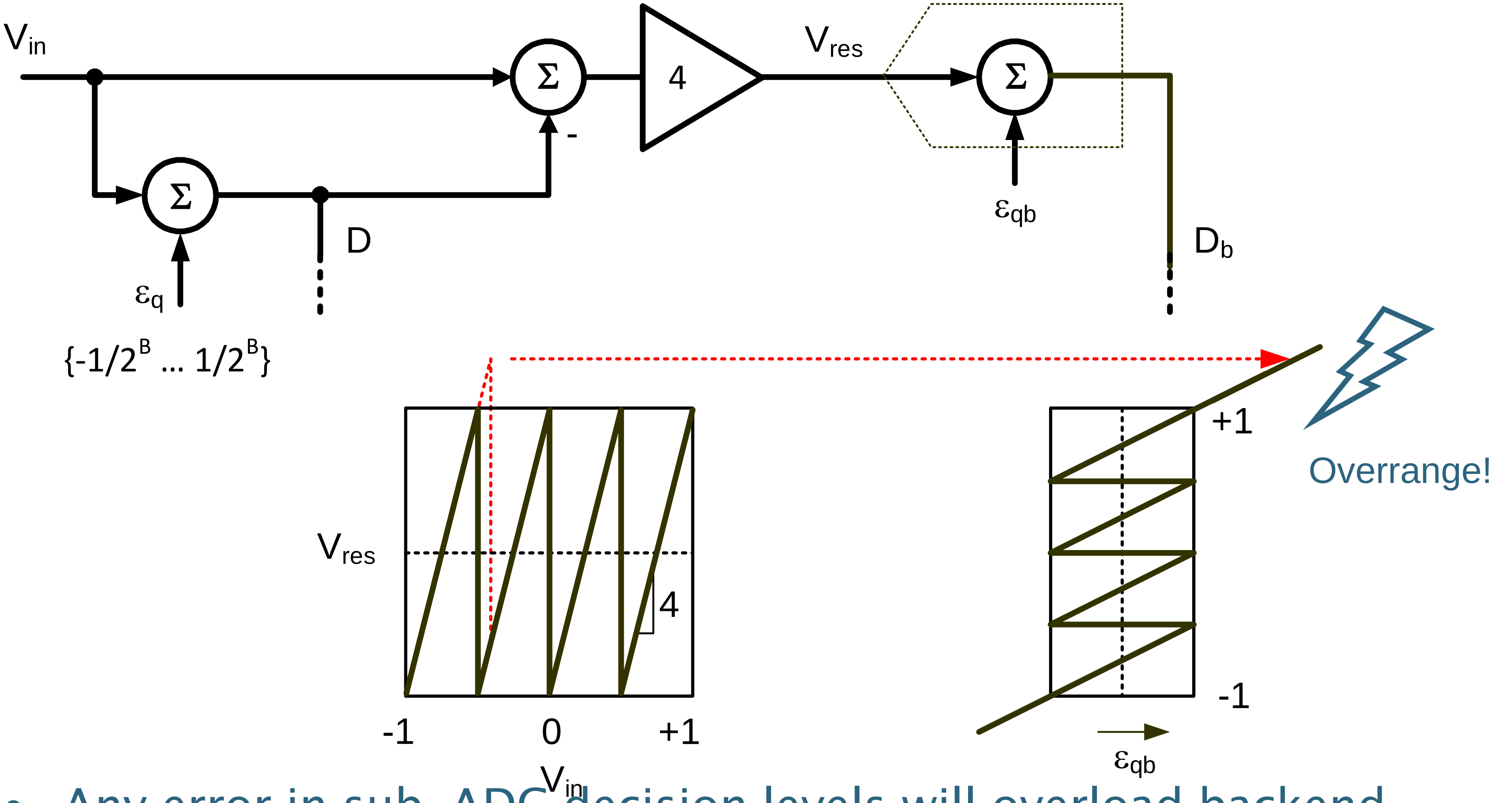
# Digital Correction



← “enlarged” residuum still within +/-2 input range of next stage

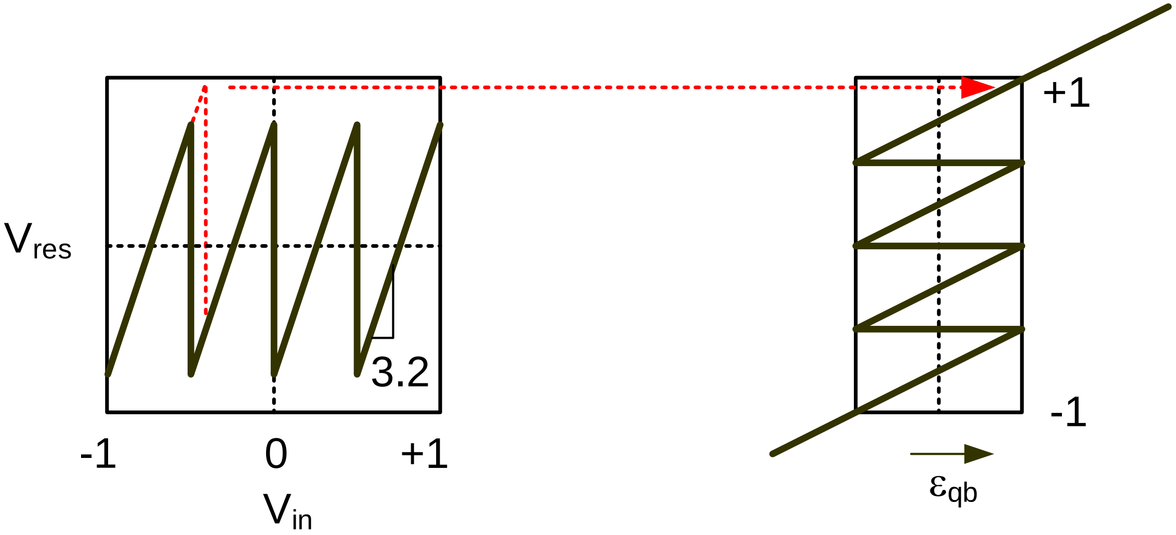
Only 1 Bit resolution from first stage (3 Bit total) →

# Issue with $G=2^B$



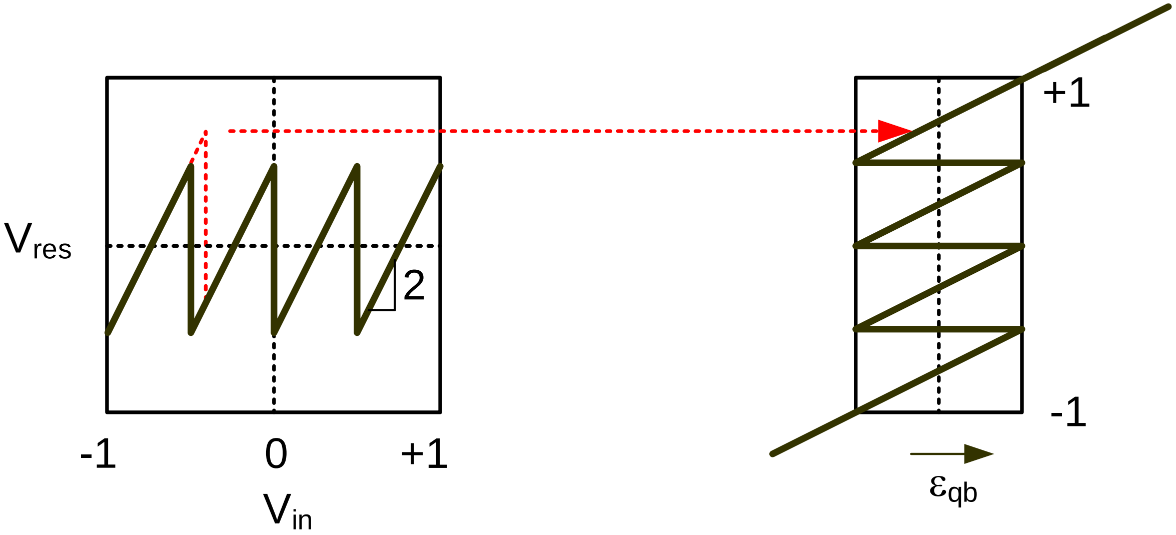
- Any error in sub-ADC decision levels will overload backend ADC and thereby deteriorate ADC transfer function

# Idea #1: G slightly less than $2^B$



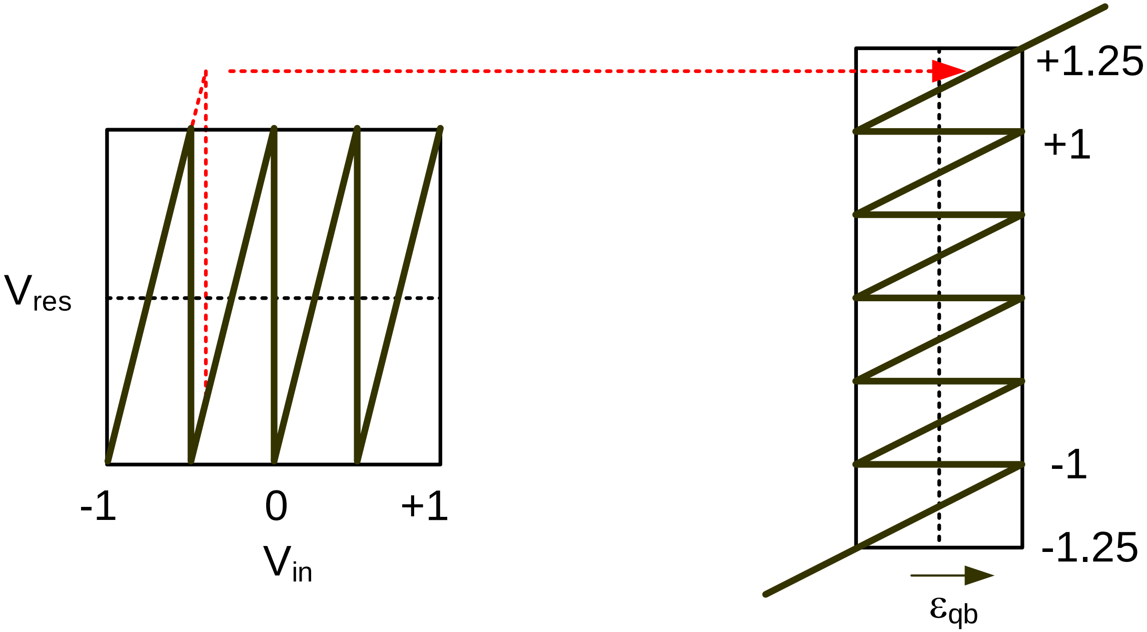
- Effective stage resolution can be non-integer ($R=\log_2 G$)
  - E.g. $R = \log_2 3.2 = 1.68$ bits
- See e.g. [Karanicolas 1993]

# Idea #2: $G < 2^B$, but Power of Two



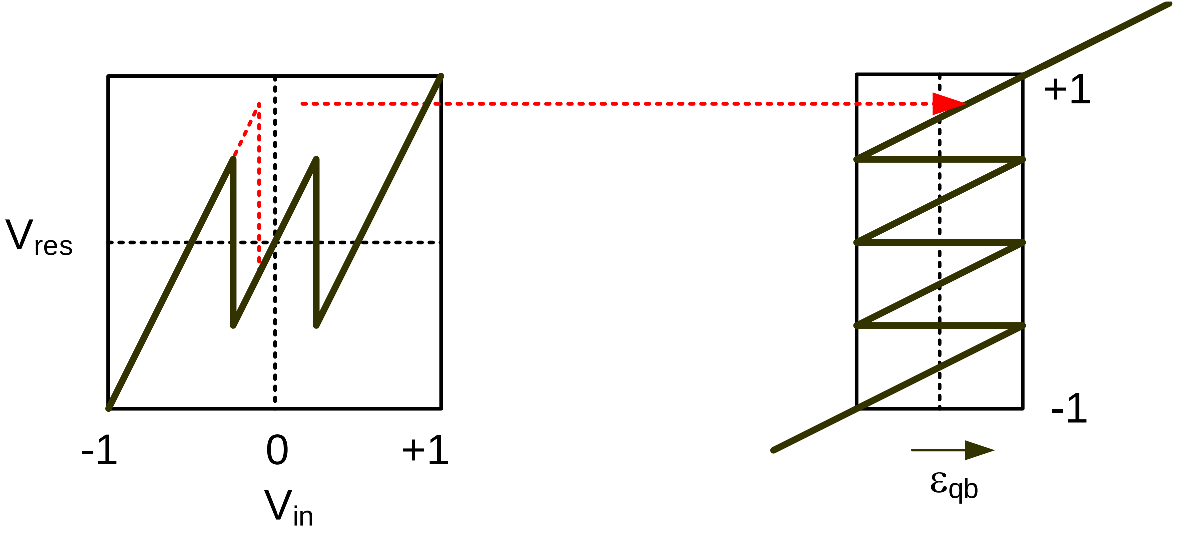
- Effective stage resolution is an integer
  - E.g. $R = \log_2 2 = 1 = B-1$
  - Digital hardware requires only a few adders, no need to implement fractional weights
- See e.g. [Mehr 2000]

# Idea #3: $G=2^B$, Extended Backend Range



- No redundancy in stage with errors
- Extra decision levels in succeeding stage used to bring residue "back into the box"
- See e.g. [Opris 1998]

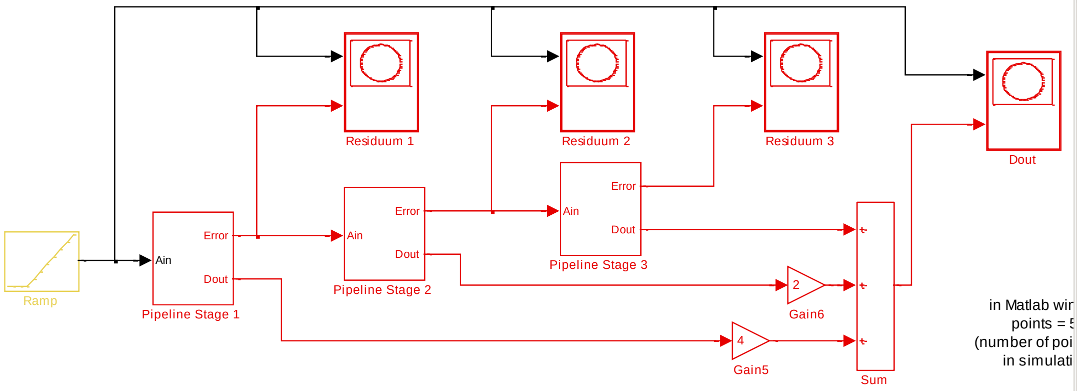
# Variant of Idea #2: "1.5-bit stage"

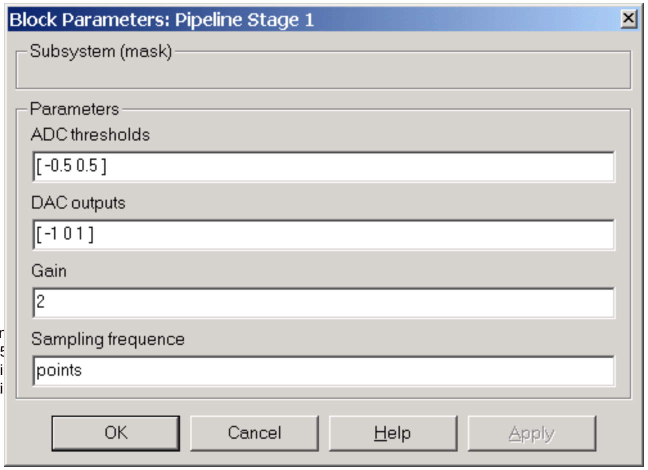


- Sub-ADC decision levels placed to minimize comparator count
- Can accommodate errors up to ±¼
- B = $\log_2(2+1)$ = 1.589 (sub-ADC resolution)
- R = $\log_2 2$ = 1 (effective stage resolution)
- See e.g. [Lewis 1992]

# “1.5-bps” Stage

Pipeline ADC,  1.5 bit per stage (2 comparators per stage)




Block Parameters: Pipeline Stage 1

Subsystem (mask)

Parameters

ADC thresholds: [-0.5 0.5 ]

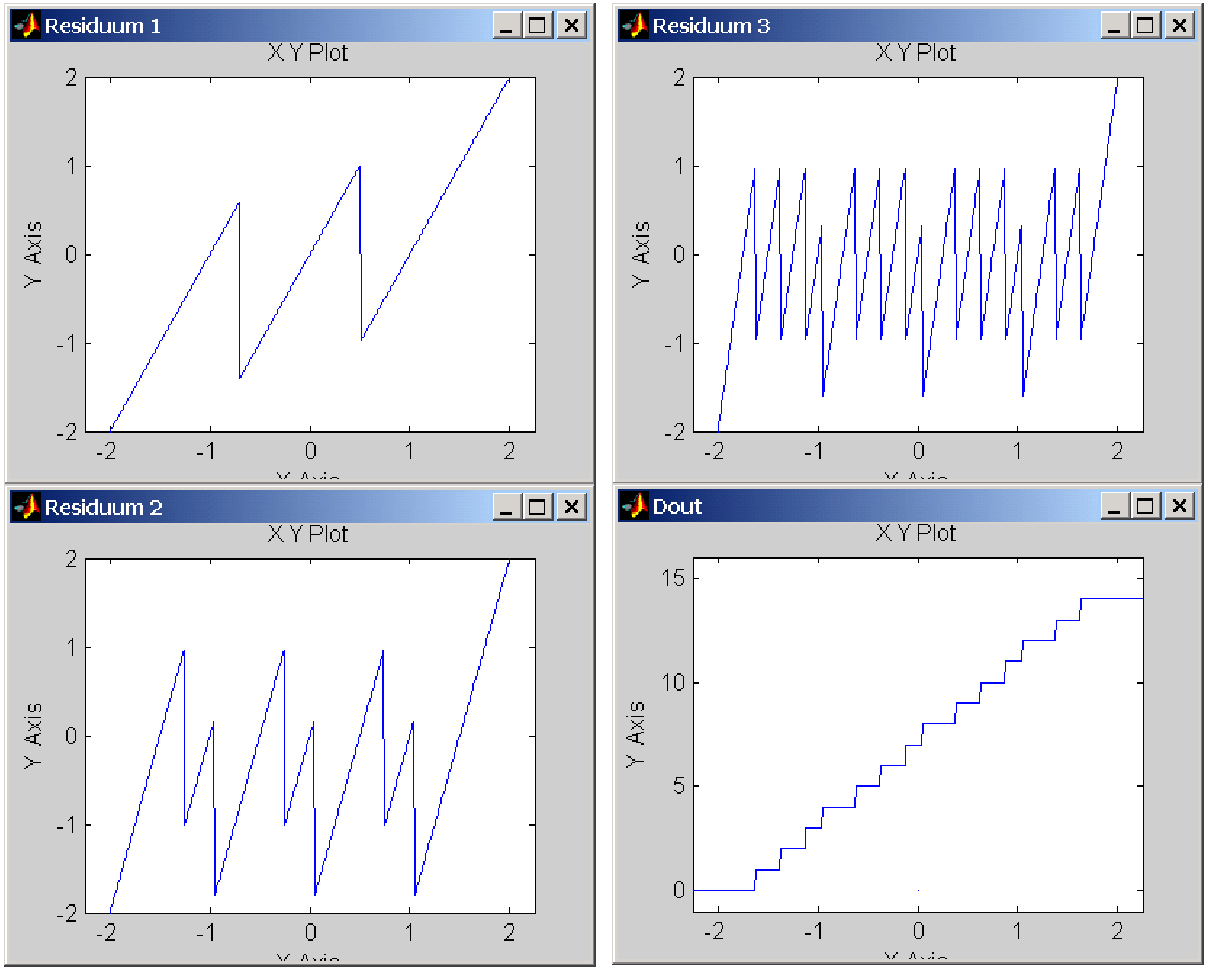
DAC outputs: [-1 0 1 ]

Gain: 2

Sampling frequence: points

- A full bit of “overrange” is excessive for typical comparator offset
  - → use only 2 (rather than 3) comparators and G=2
- 3 DAC levels → lb(3) = 1.585 Bits
- Overall resolution:
  - 1 bps for all stages but last
  - 1.585 Bit for last

# 1.5-bps Pipeline



- What is the maximum offset that can be corrected?
- What is the offset of each comparator in this example?

Ref: S. Lewis et al, “A 10-b 20-Msample/s Analog-to-Digital Converter,” J. Solid-State Circ., pp. 351-8, March 1992.

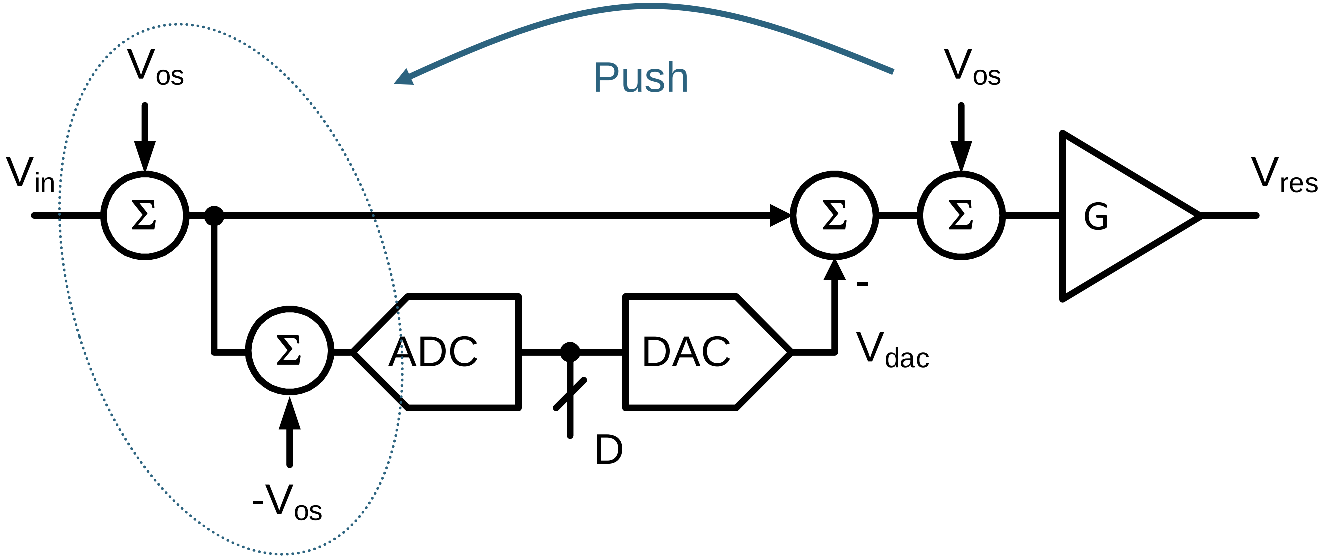
# Summary on Sub-ADC Redundancy

- We can tolerate sub-ADC errors as long as
  - The residue stays "inside the box", or
  - Another stage downstream returns the residue "into the box" before it gets clipped or reaches the last quantizer
- This result applies to any stage in an n-stage pipeline
  - Can always decompose pipeline into single stage + backend ADC
- In literature, sub-ADC redundancy schemes are often called "digital correction"
- There is no explicit error correction!
  - Sub-ADC errors are absorbed in the same way as their inherent quantization error
    - Provided there is no clipping …

# EE 240C
# Analog-Digital Interface Integrated Circuits

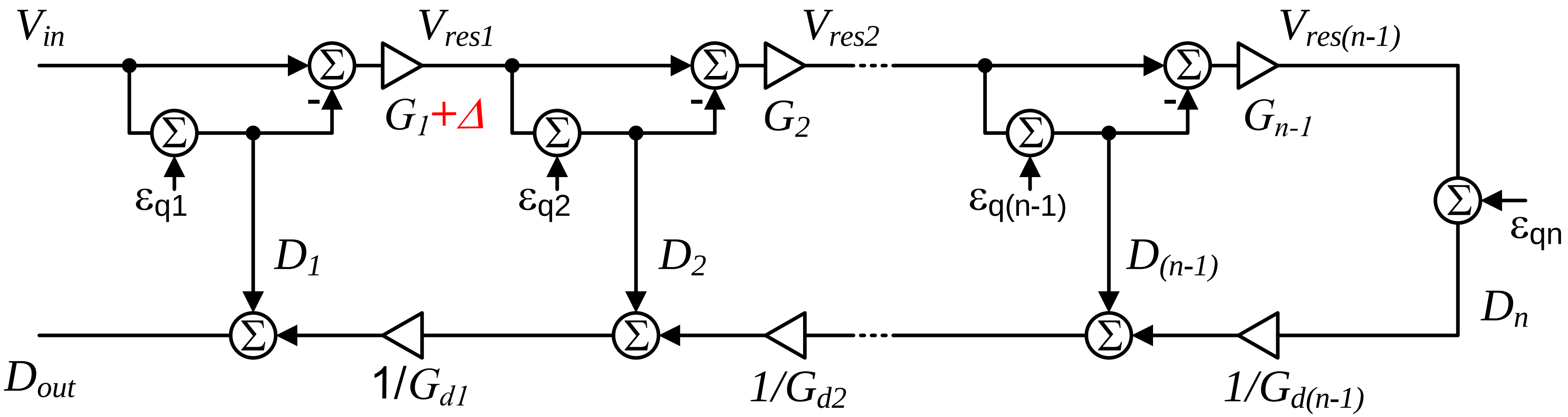
# Pipeline Amplifier Errors

# Amplifier Offset



- Amplifier offset can be referred toward stage input and results in
  - Global offset
    - Usually no problem, unless "absolute ADC accuracy" is required
  - Sub-ADC offset
    - Easily accommodated through redundancy

# Gain Errors



$$D_{out} = V_{in} + \varepsilon_{q1}\left(1 - \frac{G_1+\Delta}{G_{d1}}\right) + \ldots + \frac{\varepsilon_{qn}}{\prod_{j=1}^{n-1} G_{dj}}$$

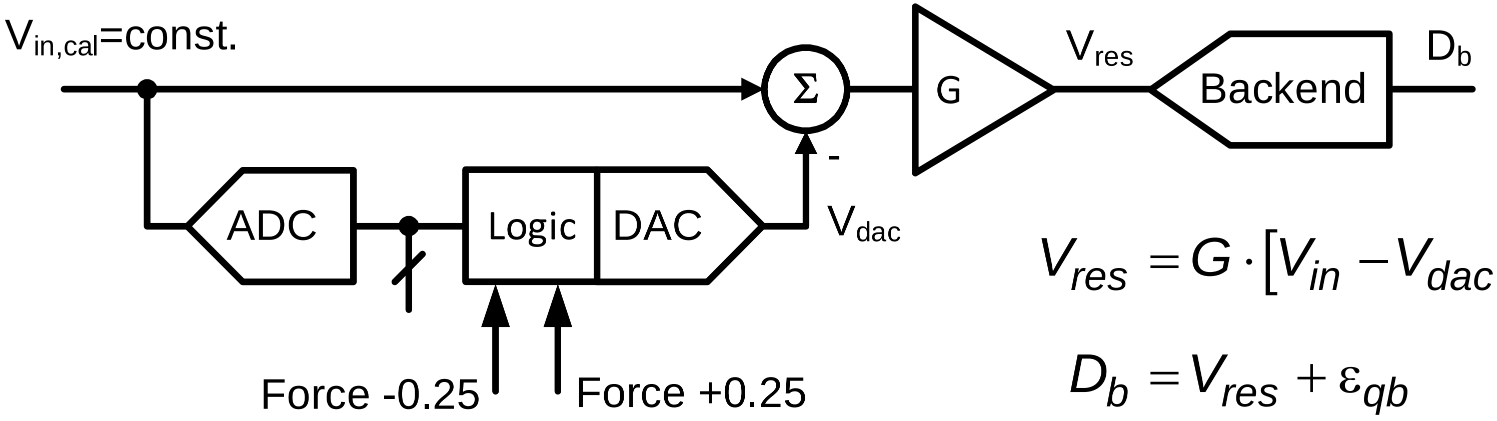
- Want to make $G_{d1} = G_1+\Delta$

# Digital Gain Calibration (1)

- Error in analog gain is not a problem as long as "digital gain term" is adjusted appropriately
- Problem
  - Need to measure analog gain precisely
- Example
  - Digital calibration of a 1-bit first stage with 1-bit redundancy (R=1, B=2)

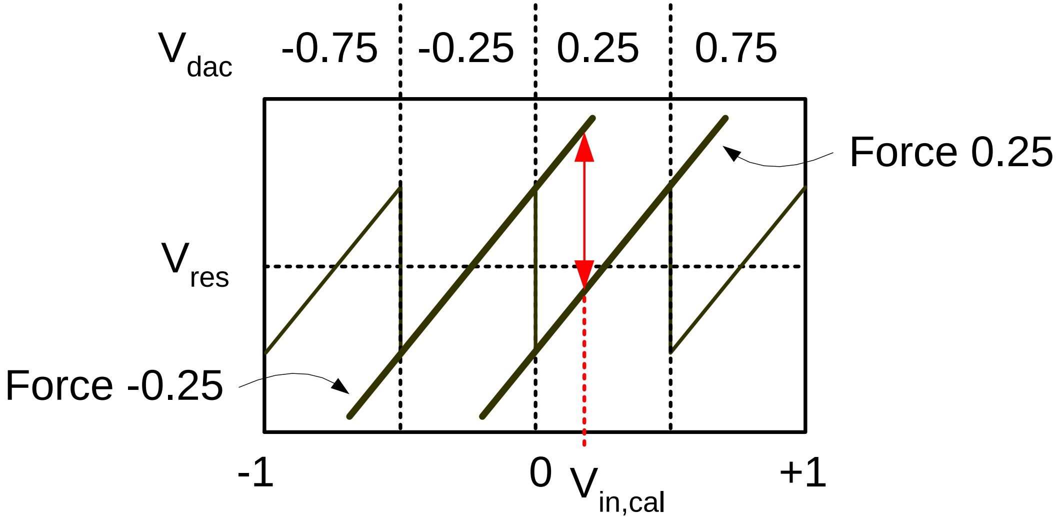
- Note
  - Even if all $G_{dj}$ are perfectly adjusted to reflect the analog gains, the ADC will have non-zero DNL and INL, bounded by ±0.5LSB. This can be explained by the fact that the residue transitions may not correspond to integer multiples of the backend-LSB. This can cause non-uniformity in the ADC transfer function (DNL, INL) and also non-monotonicity (see [Markus, 2005]).
  - In case this cannot be tolerated
    - Add redundant bits to ADC backend (after combining all bits, final result can be truncated back), or
    - Calibrate analog gain terms

# Digital Gain Calibration (2)



$$V_{res} = G \cdot [V_{in} - V_{dac}]$$

$$D_b = V_{res} + \varepsilon_{qb}$$



# Digital Gain Calibration (3)

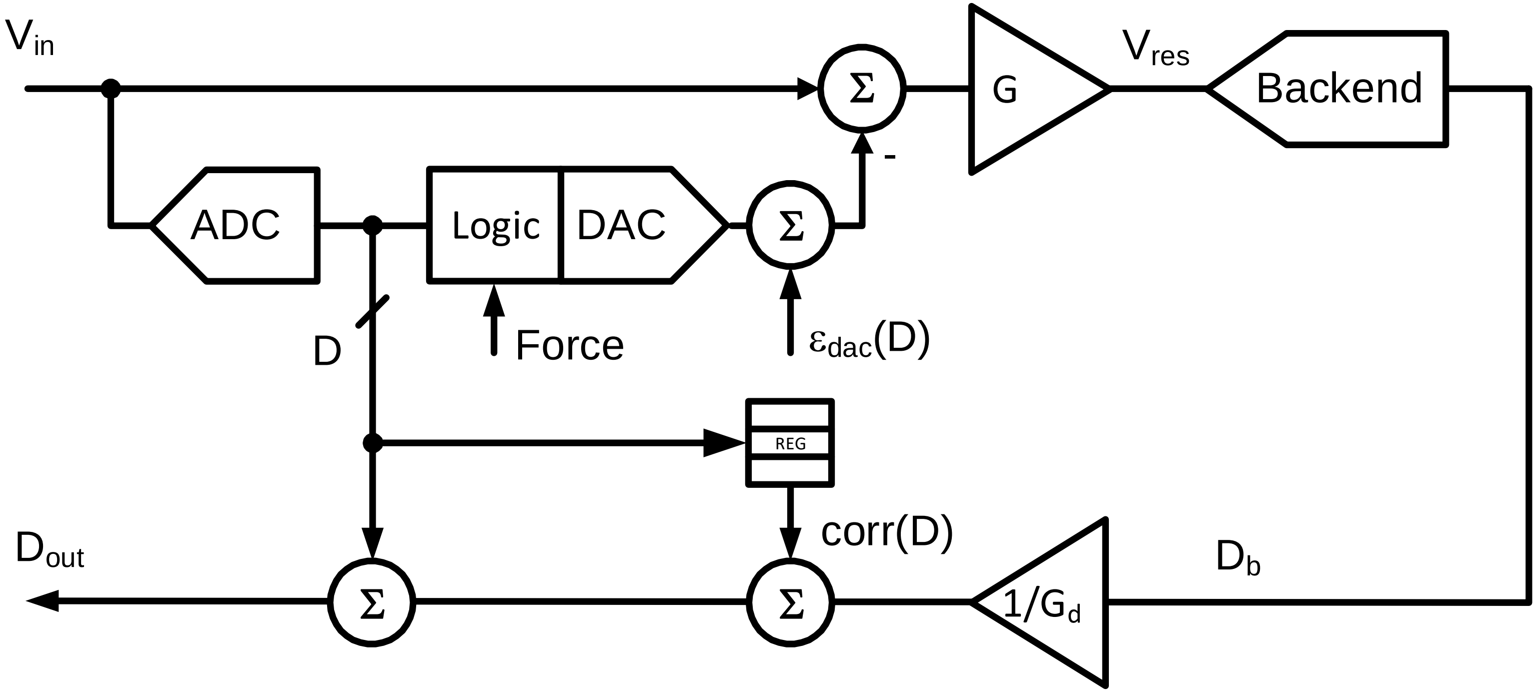
Step1: $$D_b^{(1)} = G \cdot \left[V_{in} + 0.25\right] + \varepsilon_{qb}^{(1)}$$

Step2: $$D_b^{(2)} = G \cdot \left[V_{in} - 0.25\right] + \varepsilon_{qb}^{(2)}$$

$$D_b^{(1)} - D_b^{(2)} = 0.5 \cdot G \boxed{+ \varepsilon_{qb}^{(1)} - \varepsilon_{qb}^{(2)}}$$

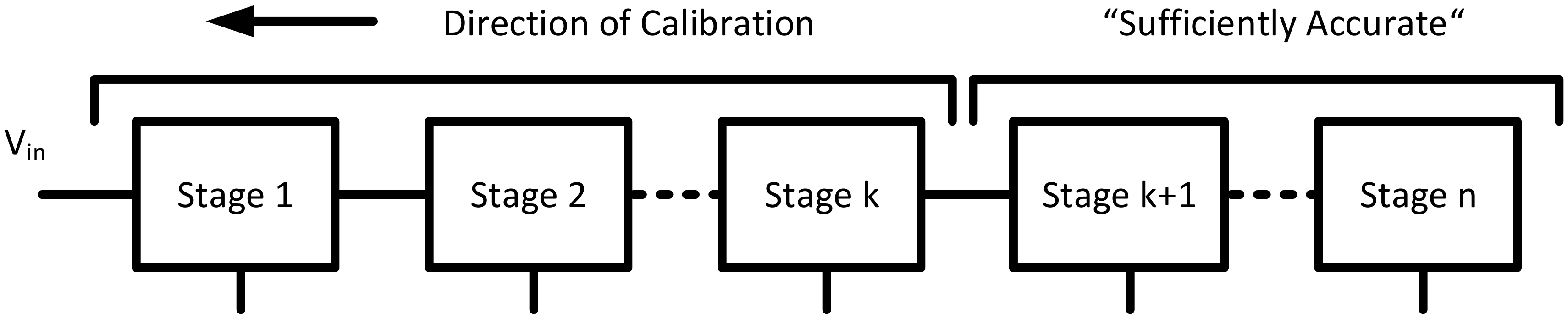
- Can minimize impact of quantization error using
  - Averaging (thermal noise dither)
  - Extra backend resolution

# DAC Calibration



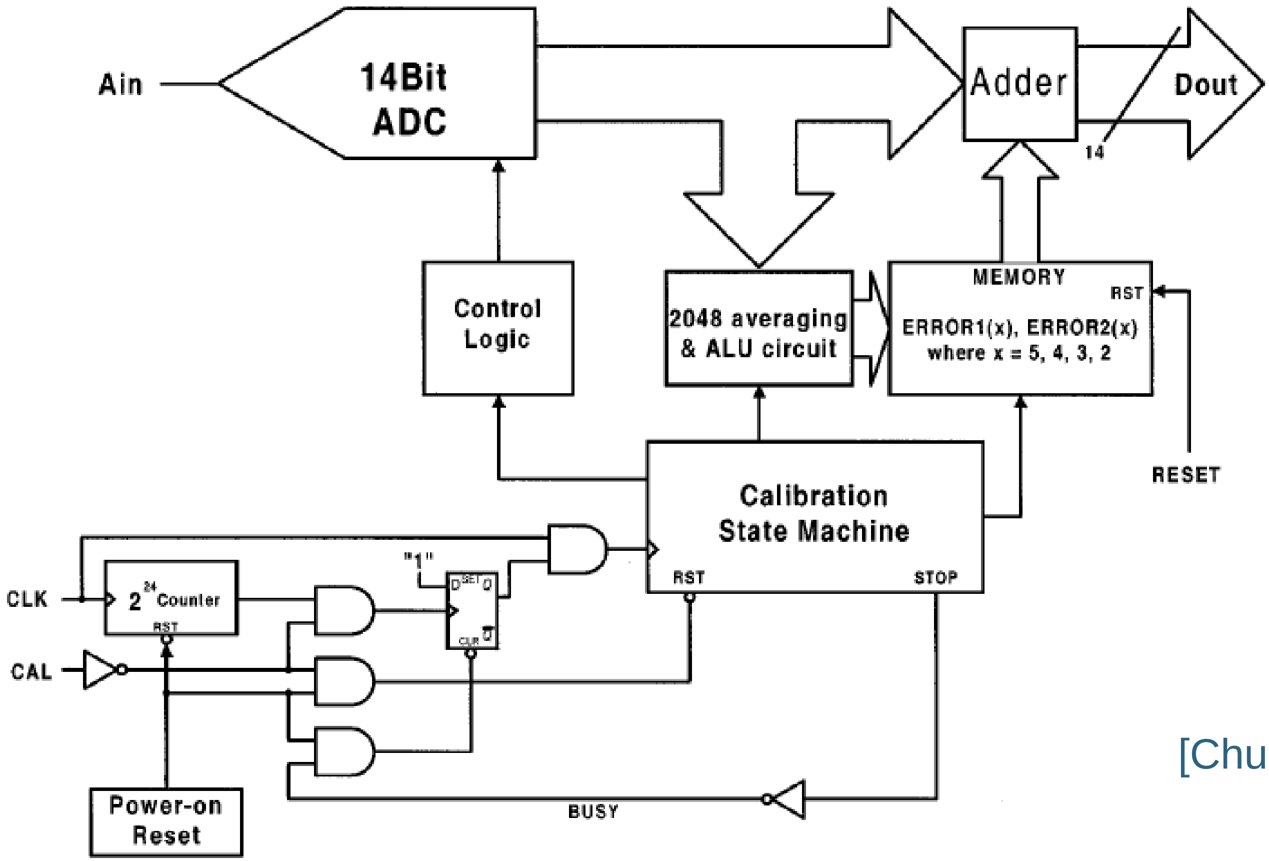
- Essentially same concept as gain calibration
  - Step through DAC codes and use backend to measure errors
- Store coefficients for each DAC transition in a look-up table

# Recursive Stage Calibration



- First few stages have most stringent accuracy requirements
  - Errors of later stages are attenuated by aggregate gain
- Commonly used algorithm [Karanicolas 1993]
  - Take ADC offline
  - Measure least significant stage that needs calibration first
  - Move to next significant stage and continue toward stage 1

# Calibration Hardware Example



[Chuang 2002]

# Alternative Schemes

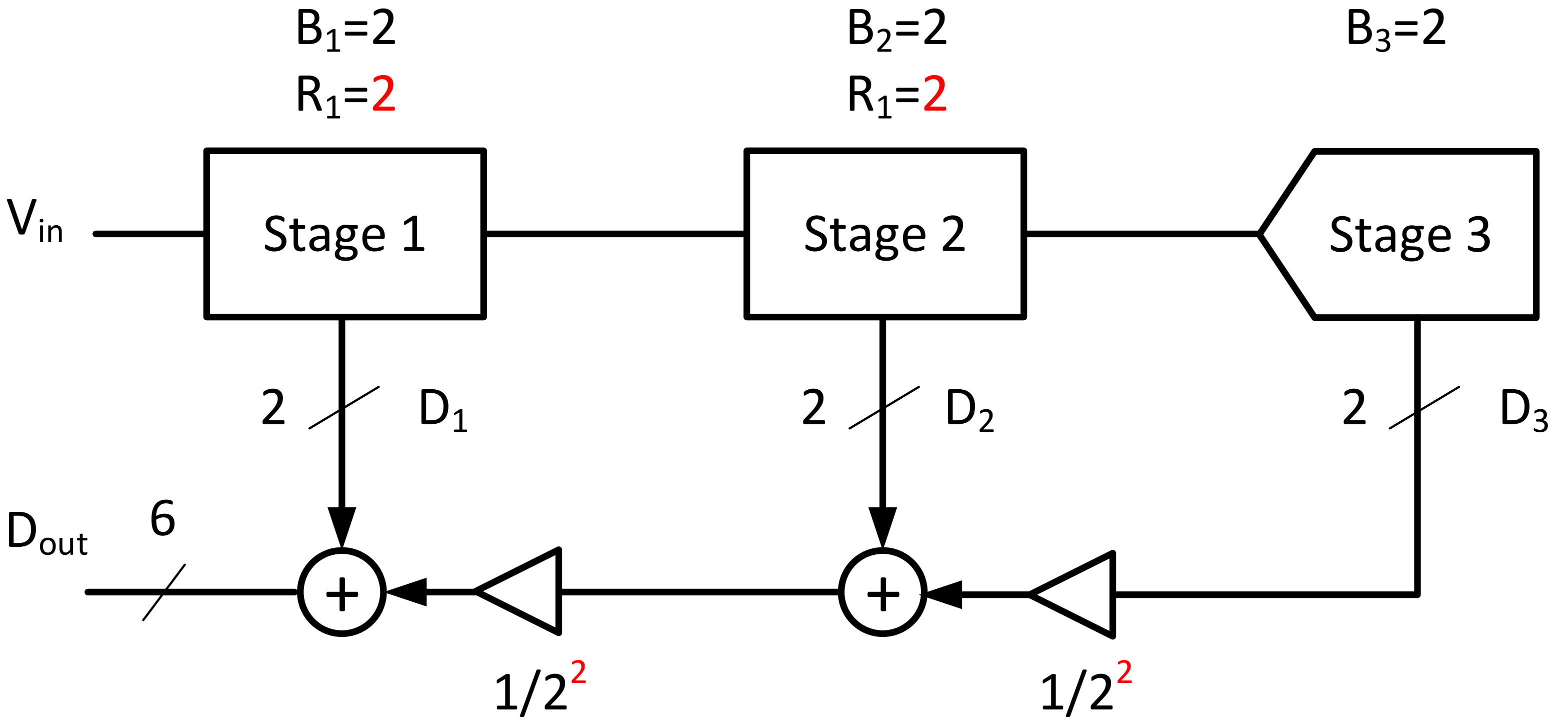
- Other foreground calibration schemes
  - Calibrate ADC starting from first stage [Singer 2000]
  - Connect stages in a circular loop [Soenen 1995]
- Background calibration
  - See e.g. [Ming 2001]
  - Makes sense primarily when calibration parameters are expected to drift
    - Capacitor ratios do not drift!
    - Background calibration is justifiable e.g. when drift in OTA open-loop gain is an issue

# EE 240C
# Analog-Digital Interface Integrated Circuits

# Pipeline Digital Backend

# Combining the Bits (1)

- Example1: Three 2-bit stages, no redundancy

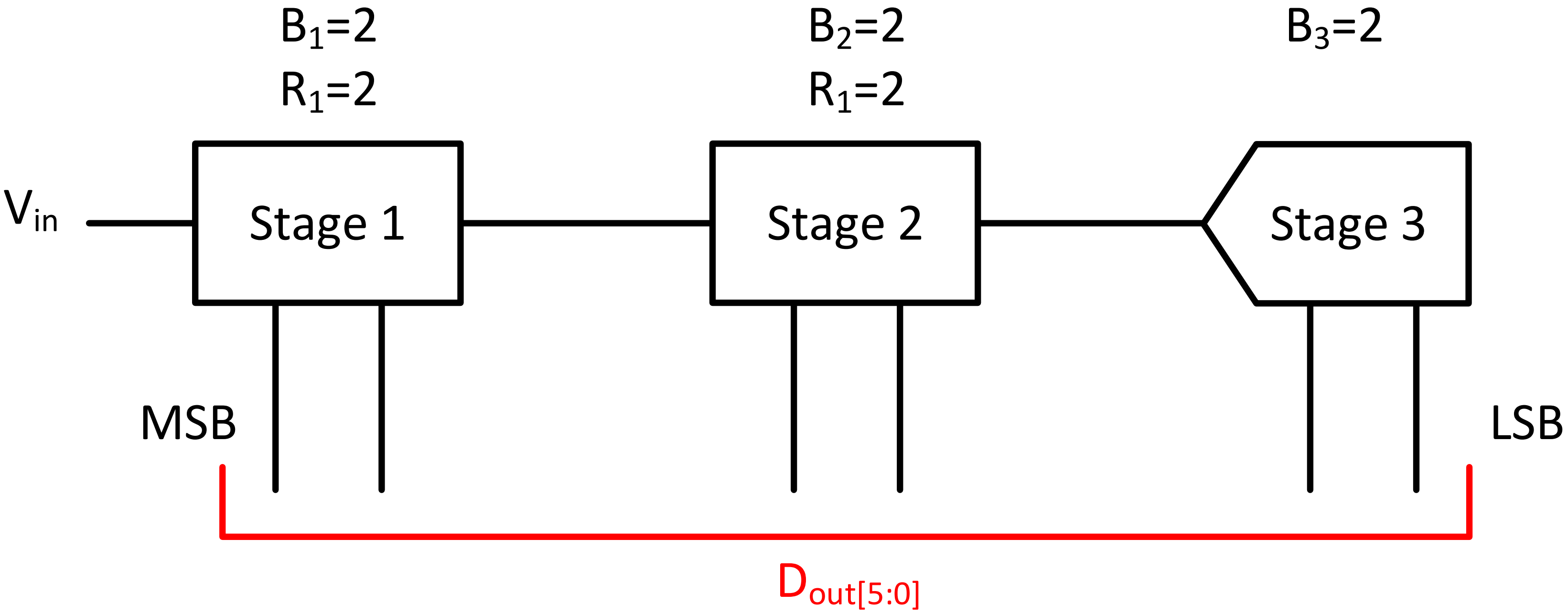


$$D_{out} = D_1 + \frac{1}{4}D_2 + \frac{1}{16}D_3$$

# Combining the Bits (2)

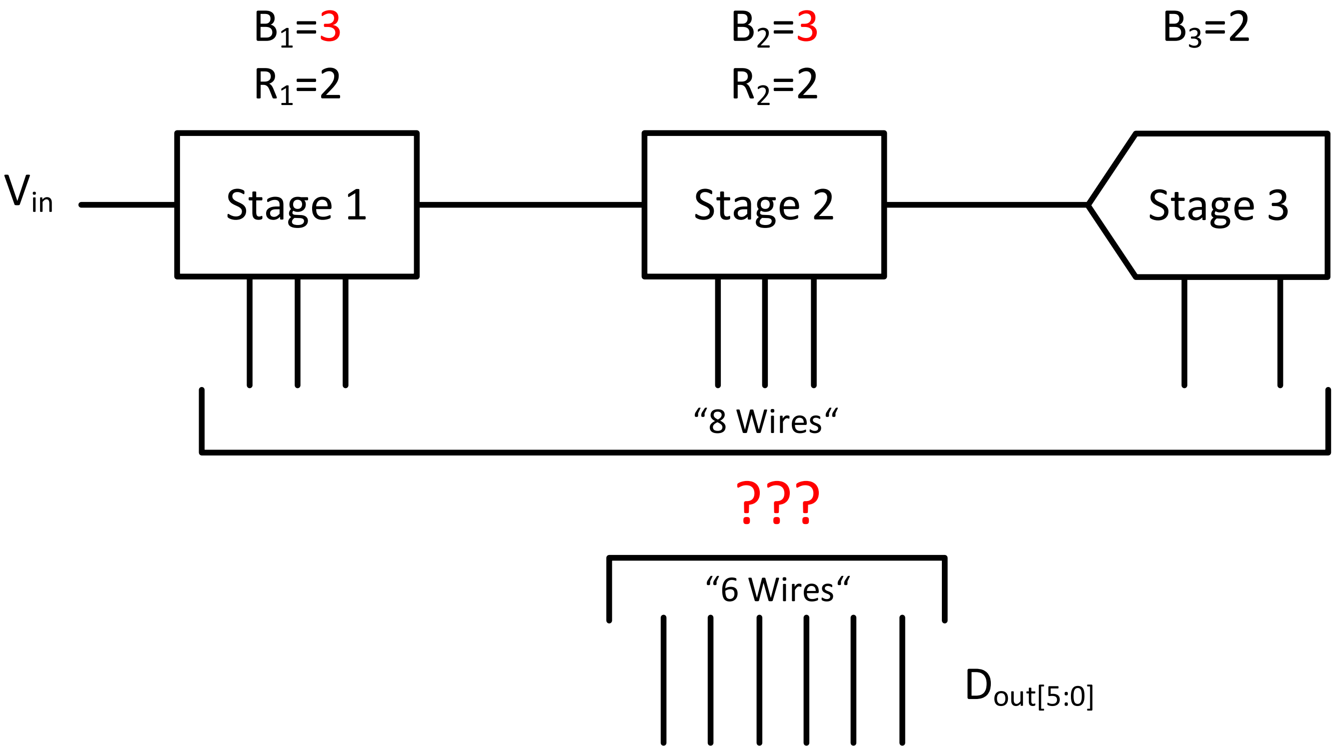
```
D_1     XX
D_2       XX
D_3         XX
-------------
D_out   DDDDDD
```

- Only bit shifts
- No arithmetic circuits needed



# Combining the Bits (3)

- Example2: Three 2-bit stages, one bit redundancy in stages 1 and 2 (6-bit aggregate ADC resolution)

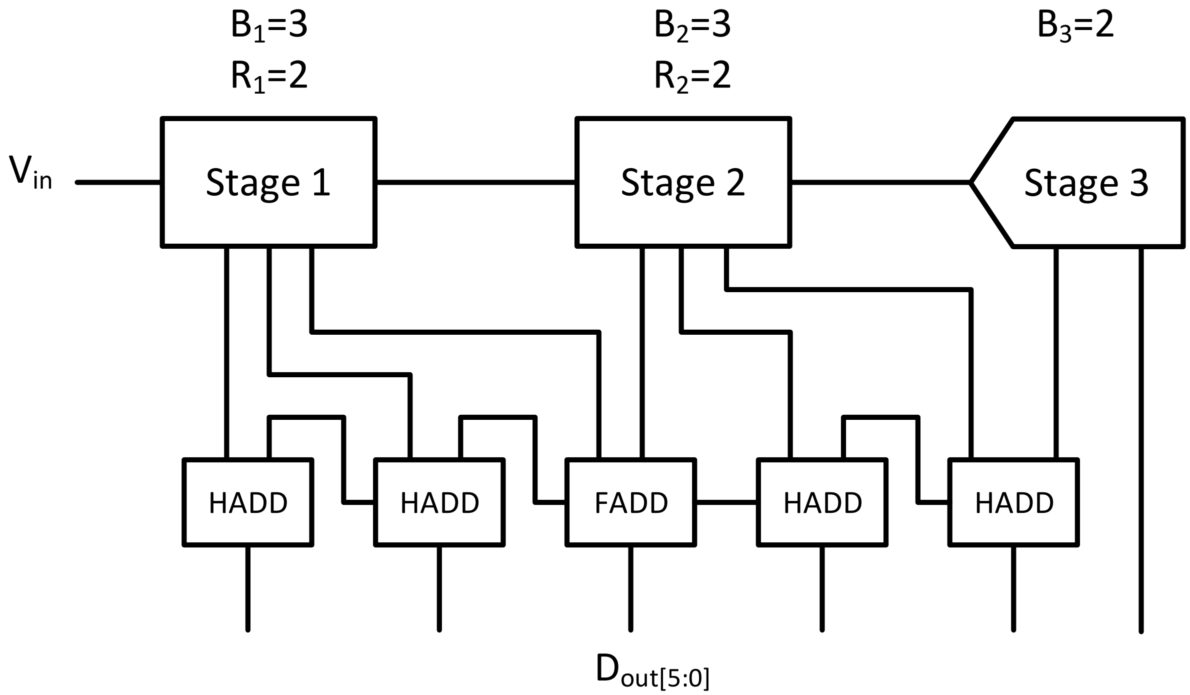


# Combining the Bits (4)

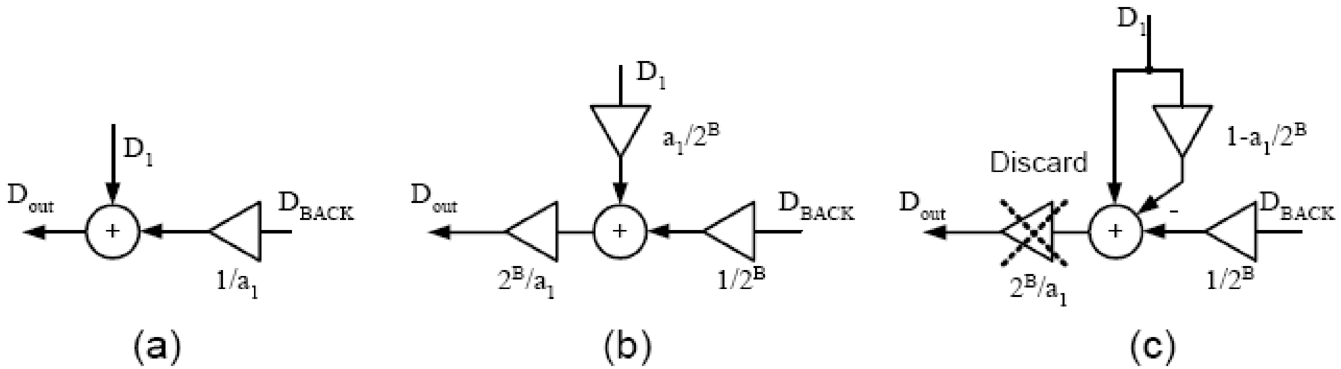
$$D_{out} = D_1 + \frac{1}{4} D_2 + \frac{1}{16} D_3$$

```
D₁     XXX
D₂        XXX
D₃           XX
------------
Dout   DDDDDD
```

- Bits overlap
- Need adders



# Combining the Bits (5)



- For fractional weights (e.g. radix <2), there is no need to implement complex multipliers
- Can still use simple bit shifts; push actual multiplication into low-resolution output
  - E.g. a 1x10 bit multiplication needs only one adder…
- See e.g. [Karanicolas 1993]