Nyquist - Rate A/D Converters

- Can be roughly divided into three categories

Low-to-Medium Speed, High Accuracy	**Medium Speed, Medium Accuracy**	**High Speed, Low-to-Medium Accuracy**
Integrating Oversampling	Successive approximation Algorithmic	Flash Two-step Interpolating Folding Pipelined Time-interleaved

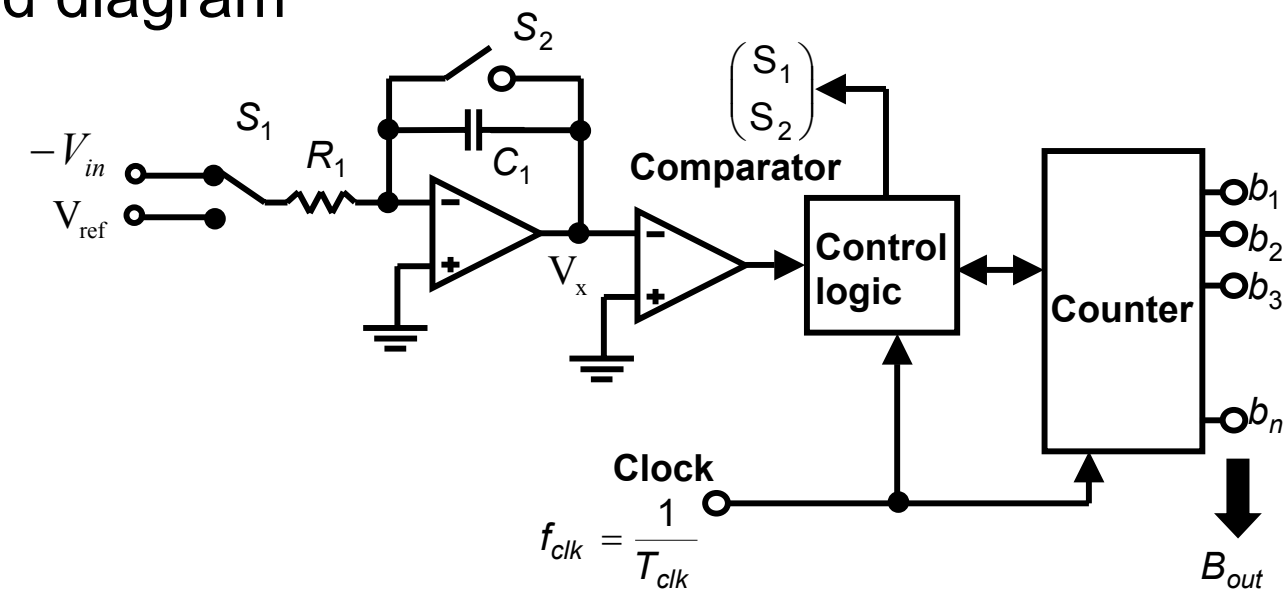
Integrating ADC(or Dual-Slope ADC)

- A popular approach for realizing high-accuracy data conversion on very slow-moving signals
- Very low offset error

  Very low gain error

  Highly linear

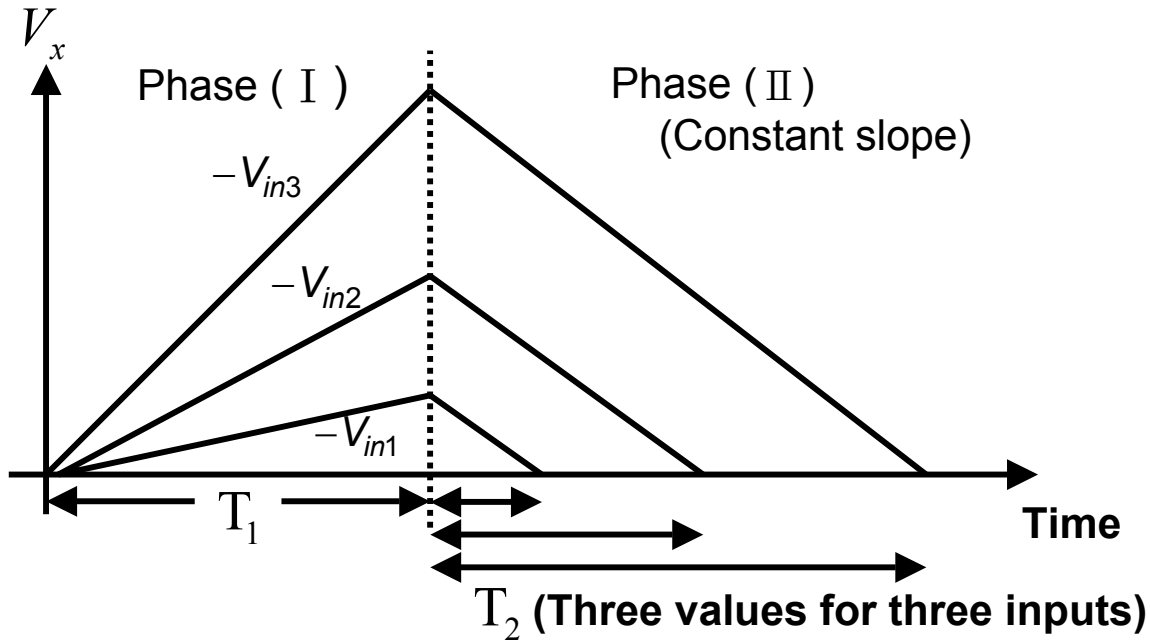
  Small amount of circuitry required
- Simplified diagram



Integrating ADC(or Dual-Slope ADC)(cont.)

- Conversion is performed in two phases
- Phase I
  - It's a fixed time interval of length T_1
    $T_1 = 2^N T_{clk}$ where T_{clk} is the period for one clock cycle
  - S_1 is connected to $-V_{in}$ such that V_x ramps up proportional to the magnitude of V_{in}
  - At the beginning, V_x is reset to zero by S_2
  - At the end of phase I, $V_x(T_1) = \frac{V_{in} T_1}{R_1 C_1}$

Integrating ADC(or Dual-Slope ADC)(cont.)



- Phase Ⅱ
  - a variable amount of time, T_2
  - At the beginning, counter is reset

    S_1 is connected to V_{ref} , resulting in a constant slope for the decaying voltage at V_x
  - The counter simply counts until V_x is less than zero

Integrating ADC(or Dual-Slope ADC)(cont.)

– Assuming the digital output count is normalized so that the largest count is unity, the counter output B_{out}, can be defined to be

$$\boldsymbol{B_{out}} = \boldsymbol{b}_1 2^{-1} + \boldsymbol{b}_2 2^{-2} + \cdots + \boldsymbol{b}_N 2^{-N}$$

and we have

$$T_2 = 2^N B_{out} T_{clk} = (b_1 2^{N-1} + b_2 2^{N-2} + \cdots + b_N) T_{clk}$$

$$V_x(t) = -\int_{T_1}^{t} \frac{V_{ref}}{R_1 C_1} d\tau + V_x(T_1) = -\frac{V_{ref}}{R_1 C_1}(t - T_1) + \frac{V_{in} T_1}{R_1 C_1}$$

Since $V_x(t) = 0$, when $t = T_{1+}T_2$

$$\frac{-\boldsymbol{V_{ref}} T_2}{\boldsymbol{R_1 C_1}} + \frac{\boldsymbol{V_{in}} T_1}{\boldsymbol{R_1 C_1}} = 0 \quad \Rightarrow \quad T_2 = T_1 \left(\frac{V_{in}}{V_{ref}}\right)$$

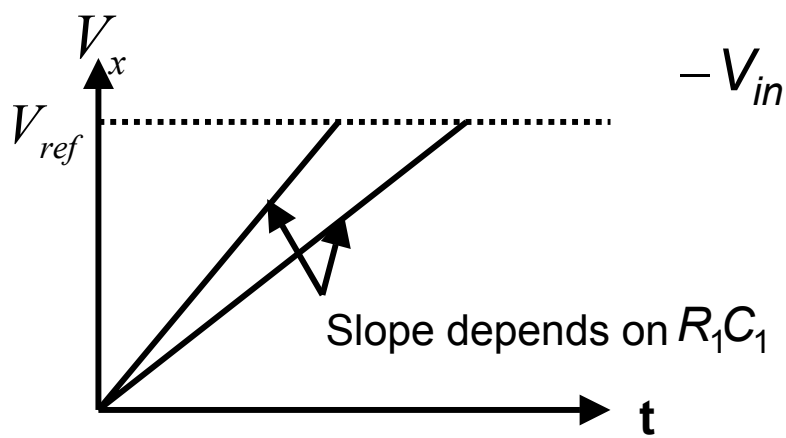
$$\Rightarrow \quad \boldsymbol{B_{out}} = \boldsymbol{b}_1 2^{-1} + \boldsymbol{b}_2 2^{-2} + \cdots + \boldsymbol{b}_N = \frac{\boldsymbol{V_{in}}}{\boldsymbol{V_{ref}}}$$

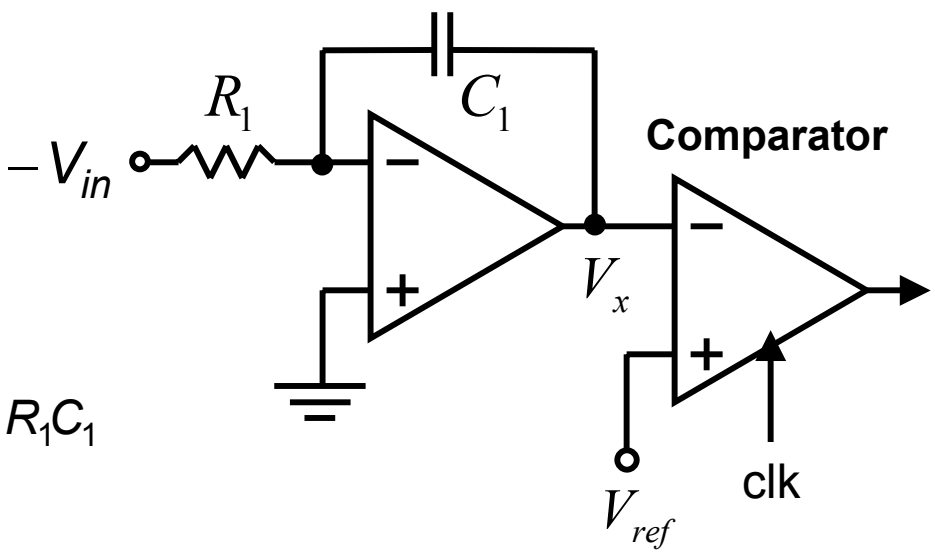
Integrating ADC(or Dual-Slope ADC)(cont.)

– From the above equations , the digital output does not depend on the time constant, R_1C_1 .

R_1 and C_1 should be chosen such that a reasonable large peak value of V_x is obtained without clipping to reduce noise effects.

– For a single-slope conversion, gain error occurs and is a function of R_1C_1 .





– To increase resolution and speed, multi-slope conversion can be used.

Integrating ADC(or Dual-Slope ADC)(cont.)

– Offset error and gain error can be calibrated
(very important mostly in DC measurement)

1. measure zero input first , then memorize its digital output, $\boldsymbol{B}_x$
2. measure full-scale DC signal, then memorize its digital output, $\boldsymbol{B}_y$

$$gain \text{ error} = (B_y - B_x) - (2^N - 1)$$

Final calibrated output $B_{out} = (B_{out} - B_x) \bullet \dfrac{2^N - 1}{B_y - B_x}$

– quite slow

$2 \bullet 2^N$ clocks are required (worse case), e.g. for a 16-bit converter with a clock frequency equal to1MHz, the worst-case conversion time is around 7.6Hz.

Integrating ADC(or Dual-Slope ADC)(cont.)

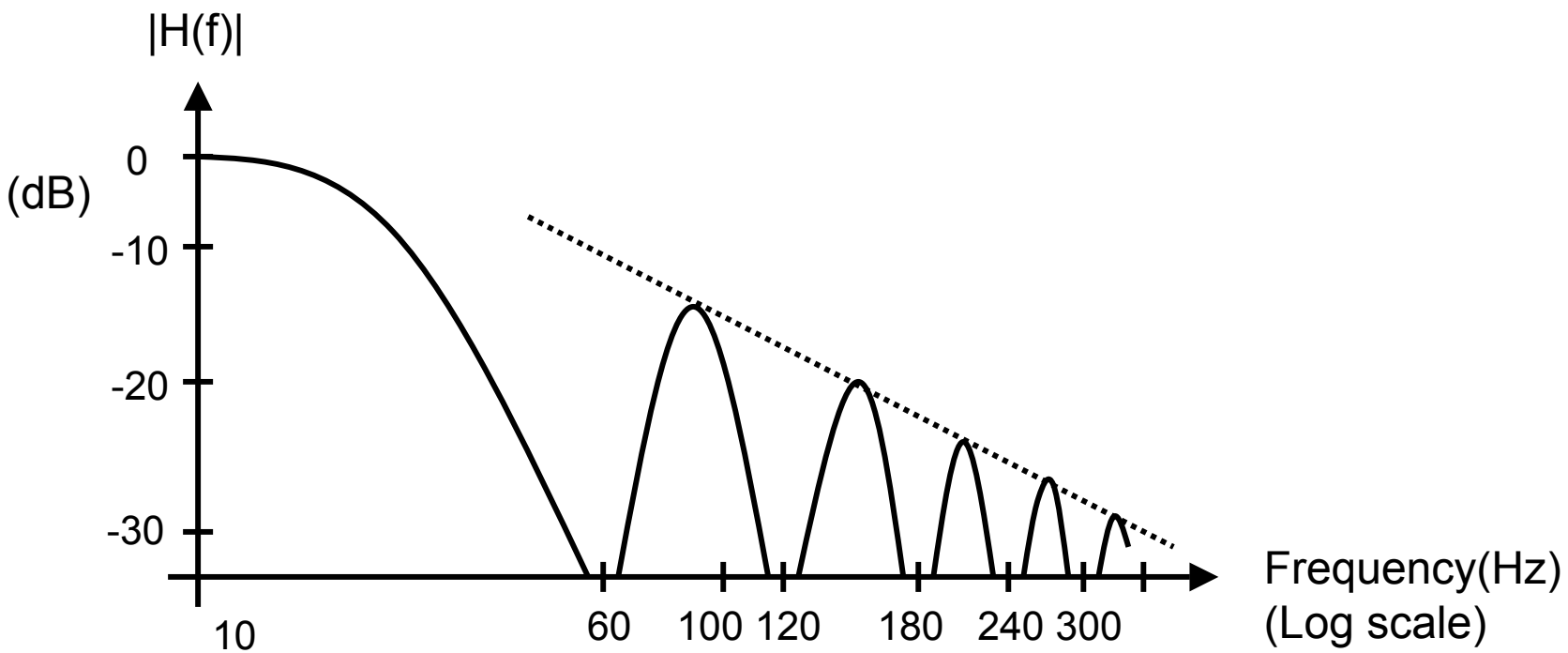
- Effective input filter with sinc function
  - By a careful choice for T_1, certain frequency components superimposed on the input signal can be significantly attenuated
  - If $\boldsymbol{v_{in}(t) = V_{in}\cos(2\pi ft)}$, where $\boldsymbol{V_{in}}$ are arbitrary magnitude

$$V_x(T_1) = -\int_0^{T_1} \frac{V_{in}\cos(2\pi ft)}{R_1C_1}dt = \frac{V_{in}}{R_1C_1} \bullet \frac{\sin(2\pi fT_1)}{2\pi f} = \frac{1}{2}H(f) \bullet \frac{V_{in}T}{R_1C_1}$$

  - Filter transfer function $\left|H(f)\right| = \left|\dfrac{\sin(\pi fT_1)}{\pi fT_1}\right|$

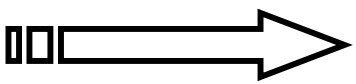
Integrating ADC(or Dual-Slope ADC)(cont.)

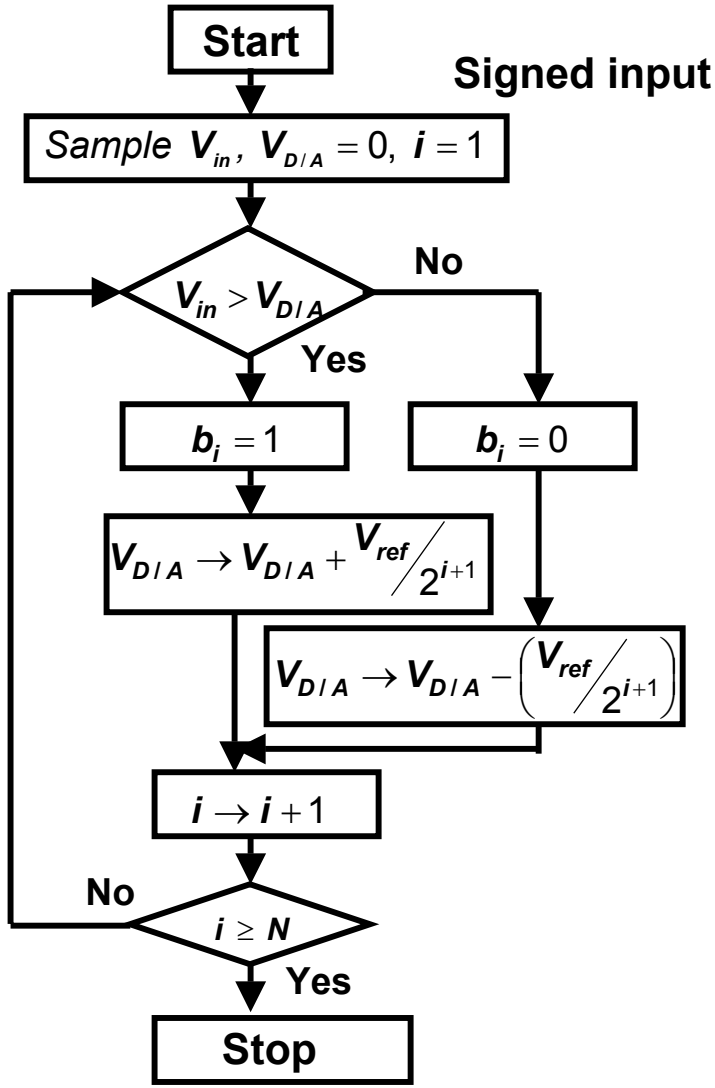
- Example
  - Filter out power line noise, especially 60Hz
    - ⇨ T_1 is equal to an integer of 16.67ms.
    - ⇨ 60Hz, 120Hz, 180Hz, …… are suppressed.



Successive Approximation (SA) ADC

- Reasonably quick conversion time
  Moderate circuit complexity
- Binary search to determine the closest digital word to match analog input,N clock cycles to complete an N-bit conversion
- Flow graph for the SA approach
  Unipolar example



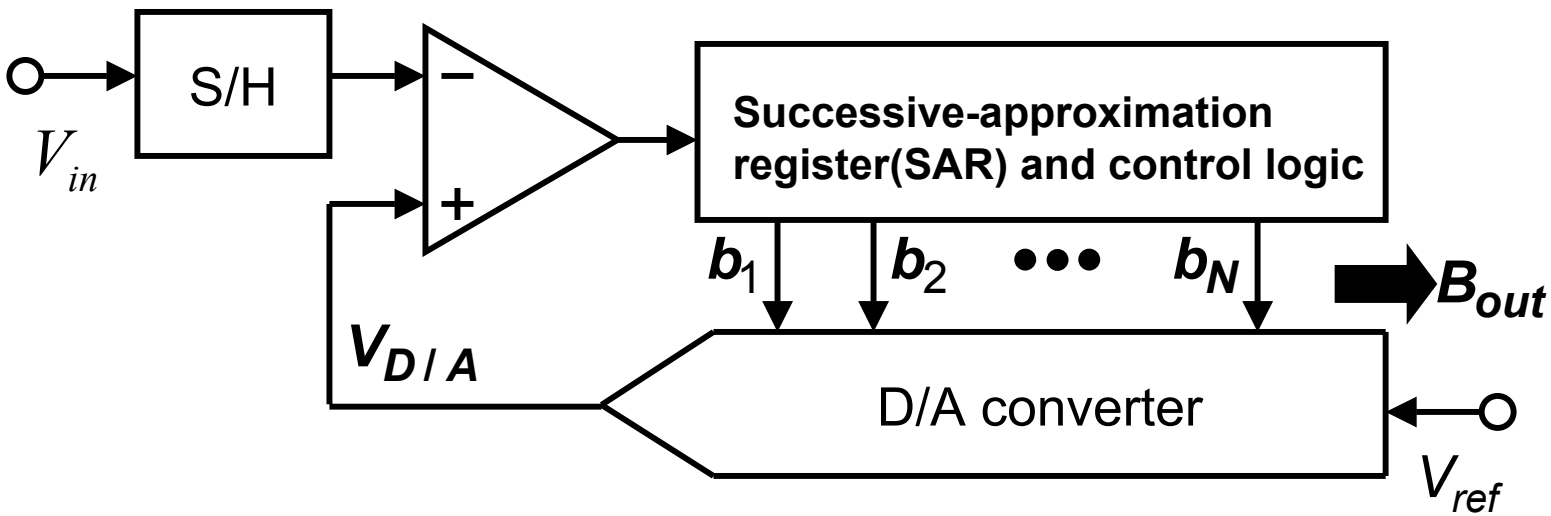


Signed input : within $\pm 0.5V_{ref}$

Unipolar output: offset-binary coding

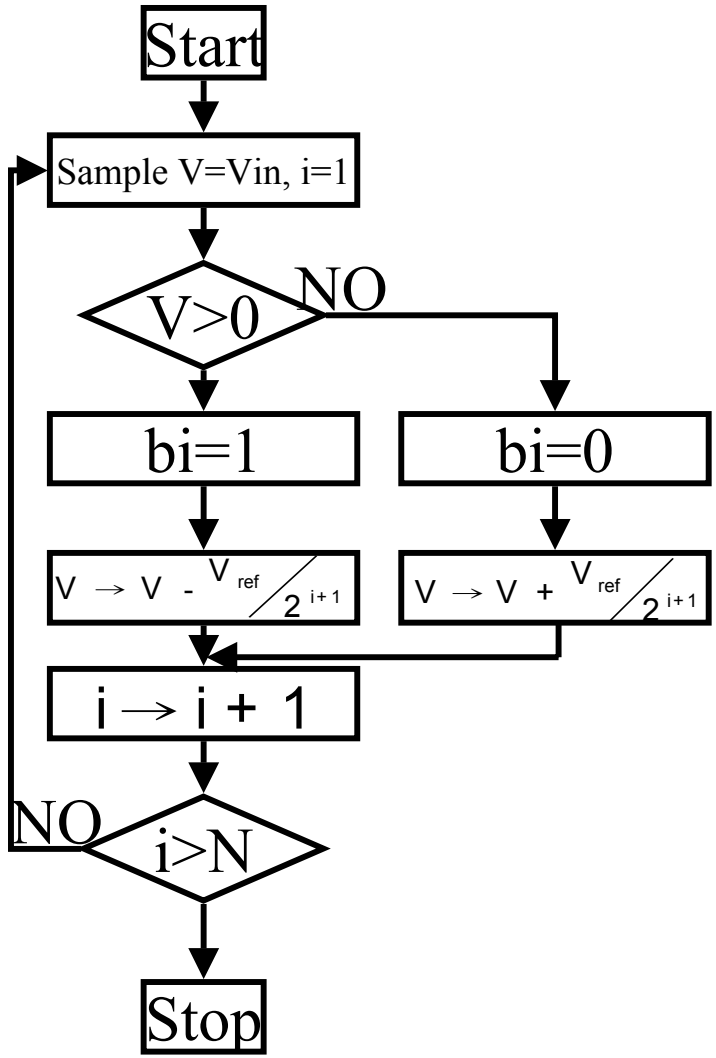
Successive Approximation (SA) ADC (Cont.)

- DAC-based successive approximation
  - unipolar example shown below (using offset-binary coding)
  - internal DAC typically determines the accuracy and sped of the SA ADC
  - sample/hold is required so that input does not change during the conversion time.



Unipolar Charge-Redistribution ADC

- Flow graph modified from that in the previous page
  - unipolar : signed input and offset binary output
  - no need of a separate DAC
    1. the error V equals the difference between input V_{in} and DAC output.
    2. V is always compared to ground.
    3. Charge-redistribution MOS. ADC is one of the first switched-capacitor ADC using this approach.



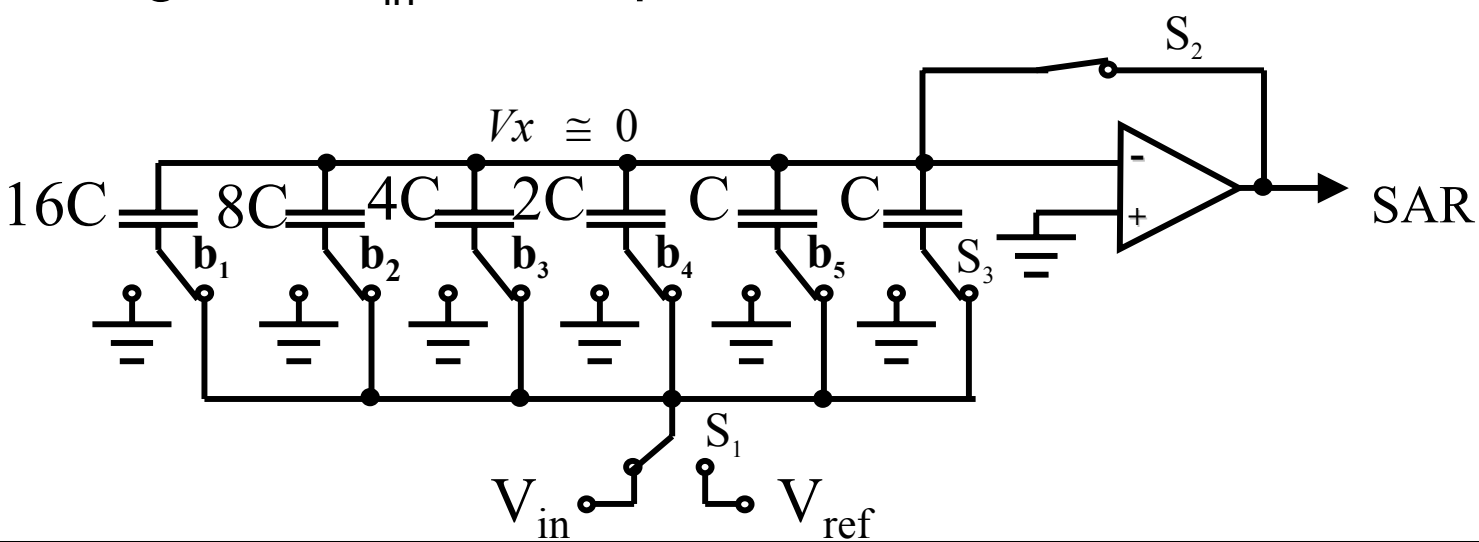
Unipolar Charge-Redistribution ADC (cont.)

4.S/H, DAC, and difference portion of the comparator are all combined into a single circuit.

- Example : A 5-bit ADC
  - 3 operational modes

    1. sample mode

       Comparator is reset though S_2. All capacitors are charged to V_{in}, which performs S/H



Unipolar Charge-Redistribution ADC (cont.)

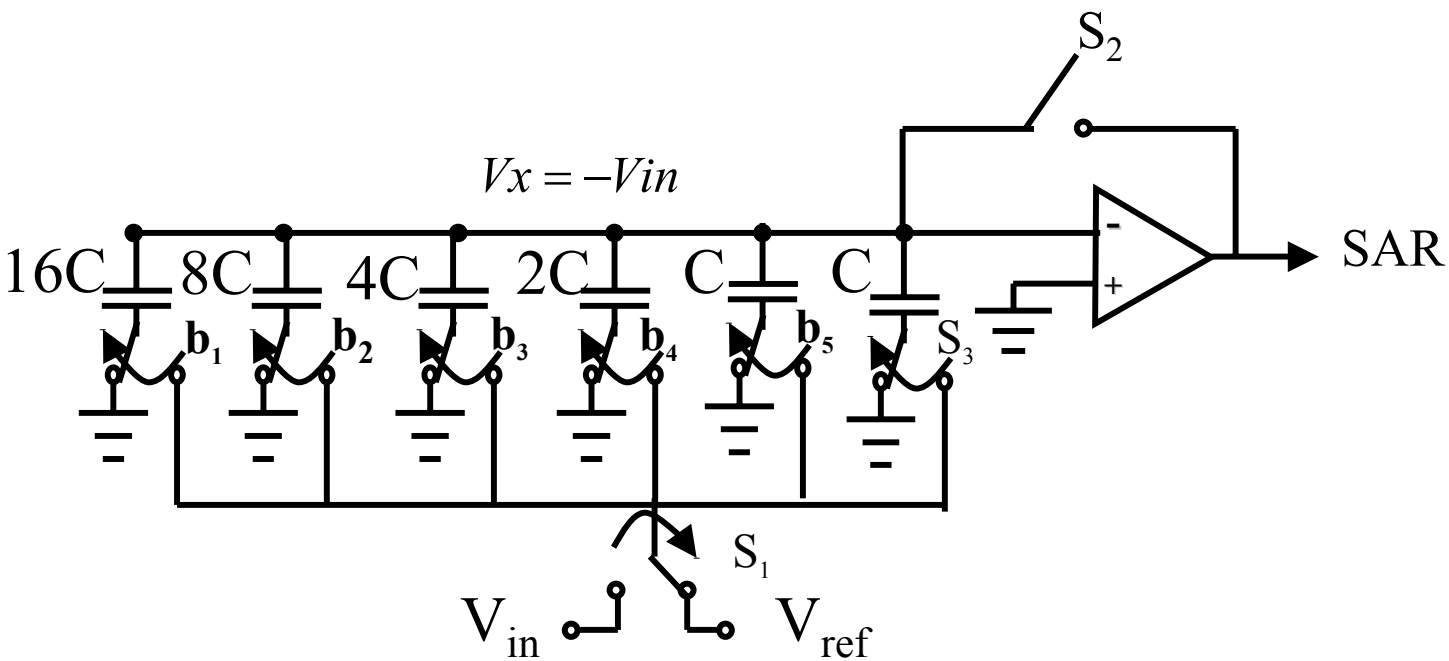
2. Hold mode

Comparator is taken out of reset.

All capacitors are switched to ground.

V_x: 0→-V_{in}

Vin is held on the capacitor array



Unipolar Charge-Redistribution ADC (cont.)

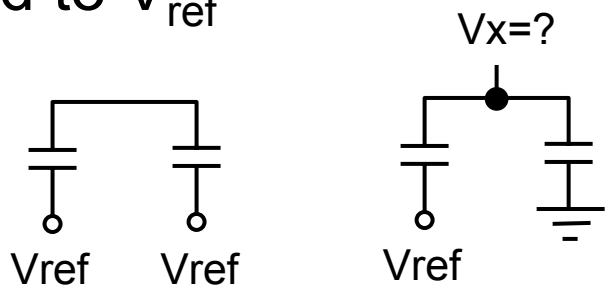
3. Bit cycling

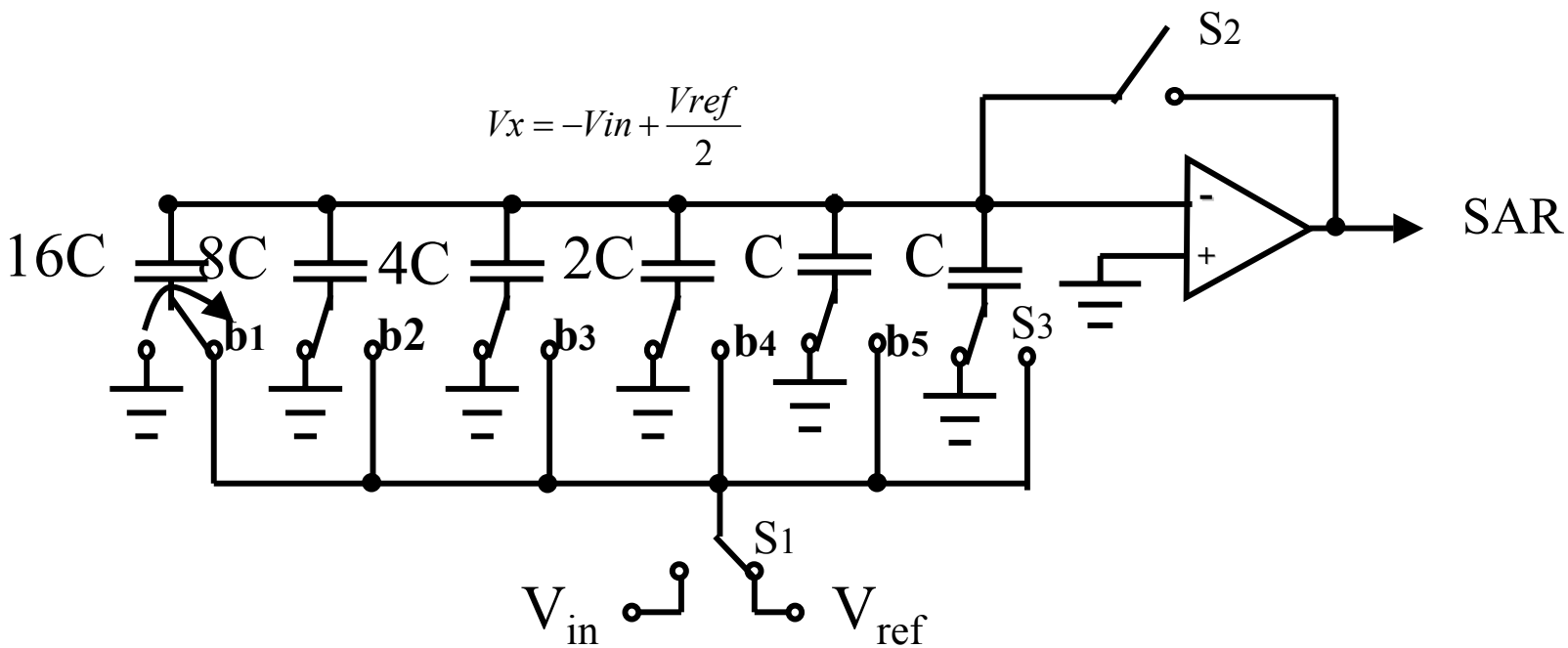
the largest capacitor is switched to V_{ref}

$V_x=-V_{in}+V_{ref}/2$

⋮

Conversion is finished





Unipolar Charge-Redistribution ADC (cont.)

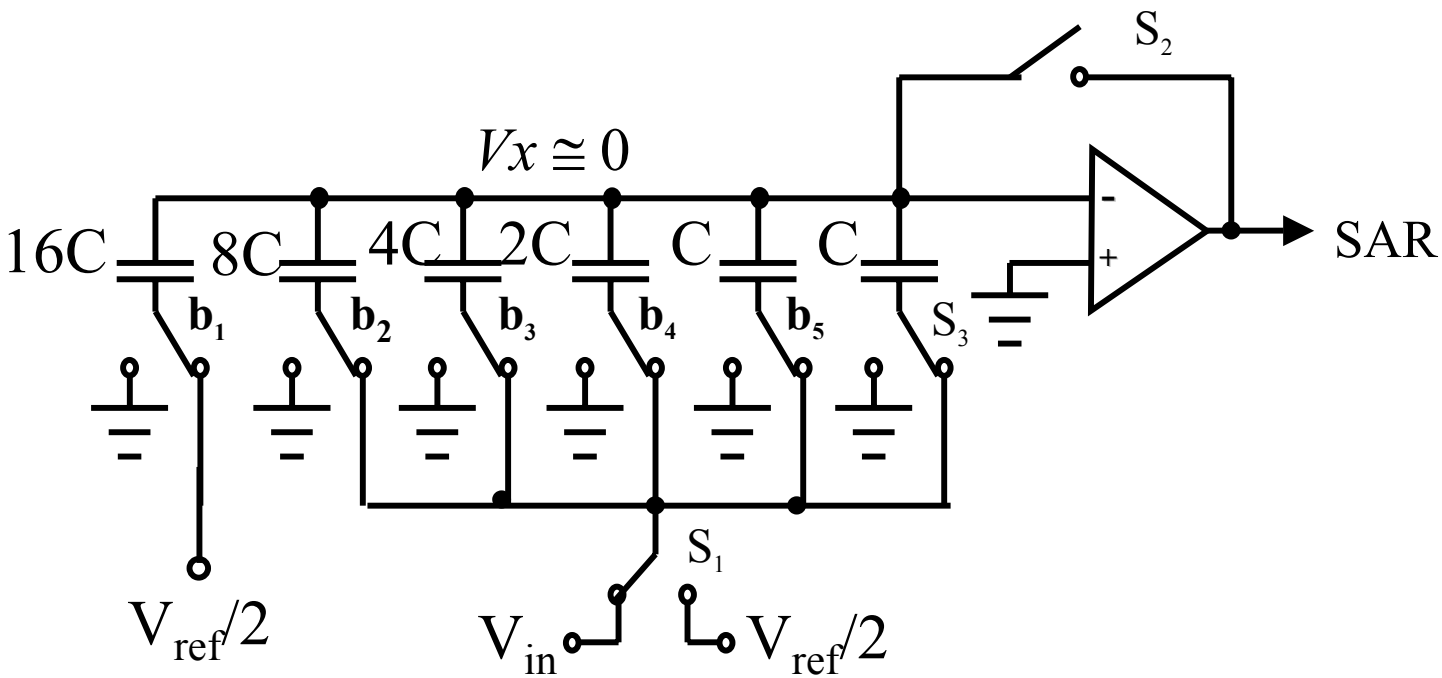
- To get an exact division by two, an additional unit capacitor is added so that the total capacitance is $2^N C$ rather than $(2^N-1)C$
- Capacitor bottom plats should be connected to V_{ref}….. It does attenuate V_x.

Signed charge-Redistribution ADC with a Single Reference Voltage

- A signed A/D conversion can be realized by adding a $-V_{ref}/2$ input. If V_x is less than zero at the first step, then proceed as in the unipolar case using V_{ref}. Otherwise, if V_x is greater than zero, use $-V_{ref}$ and test for V_x greater than zero when deciding whether to leave the capacitors connected to $-V_{ref}$ or not at each bit cycling.
- The same structure as the unipolar case can be used to realize a signed A/D conversion while using only a single V_{ref} if slightly modified switching arrangement is used.
  - A 5-bit example (V_{in} is between $\pm V_{ref}/2$)

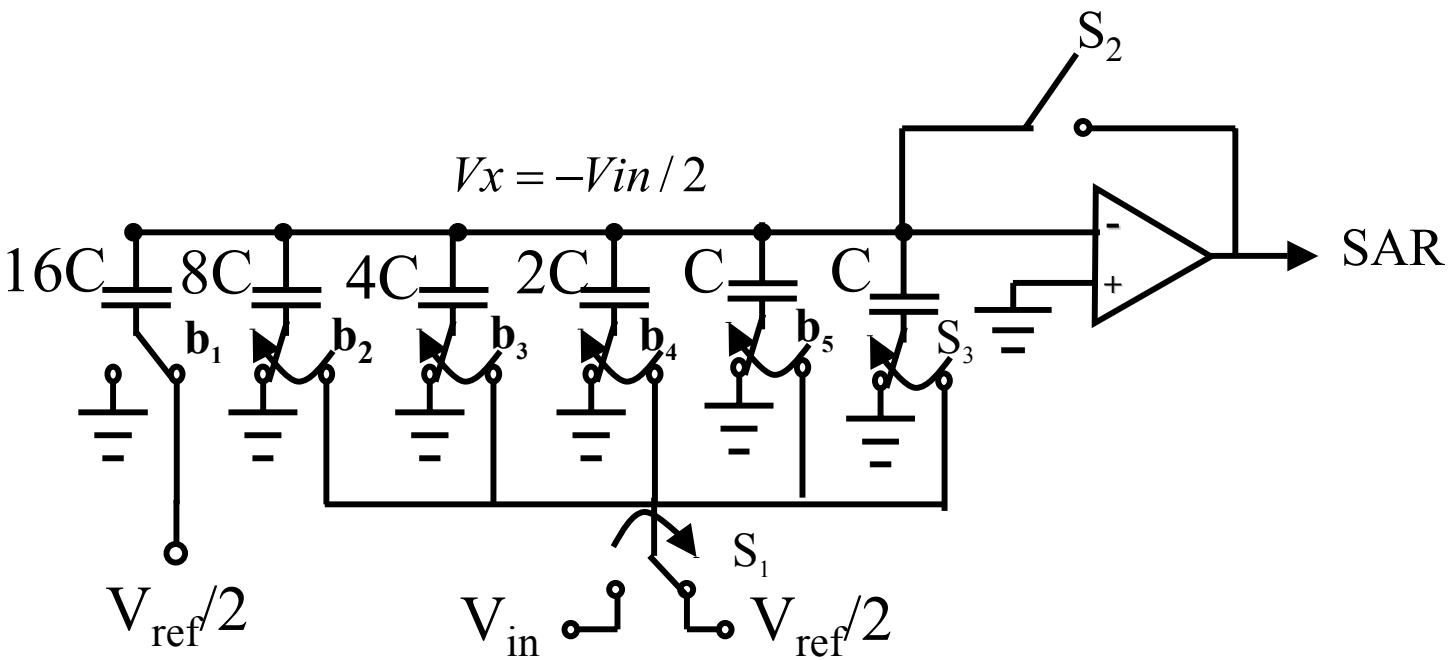
Signed charge-Redistribution ADC with a Single Reference Voltage (cont.)

1. Sample mode: in the first step, all the capacitors, except for the largest capacitor, are charged to V_{in} while the comparator is being reset to its threshold voltage. For the signed case, the largest is now connected to $V_{ref}/2$.



Signed charge-Redistribution ADC with a Single Reference Voltage (cont.)

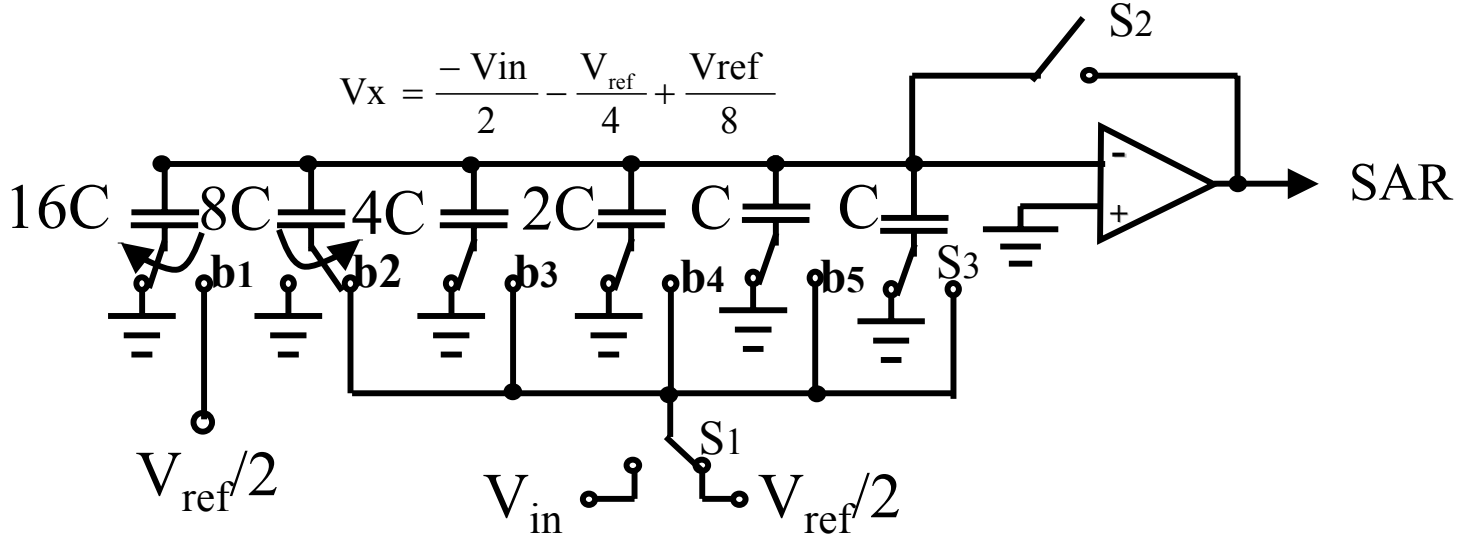
2. Hold mode: Next, the comparator is first taken out of reset, and then all the capacitors, except the largest one, are switched to ground. This causes V_x, which was originally zero, to change to $-V_{in}/2$. At the end of this step, the sign of the input signal is determined by looking at the comparator output.



Signed charge-Redistribution ADC with a Single Reference Voltage (cont.)

3. Bit cycling: Next, the largest capacitor(i.e., the 16C capacitor in this example)is switched to ground if and only if , V_x is larger than zero (i.e., when V_{in} is less than zero). Specifically, if V_x is larger than zero then V_{in} is positive and b1 is set to 0, the largest capacitor is switched to ground, causing V_x to become $-V_{in}/2-V_{ref}/4$(which is a negative value), and conversion proceeds as in the unipolar case, starting with b_2. Once conversion is completed, some digital recording may be required to obtain the desired output code

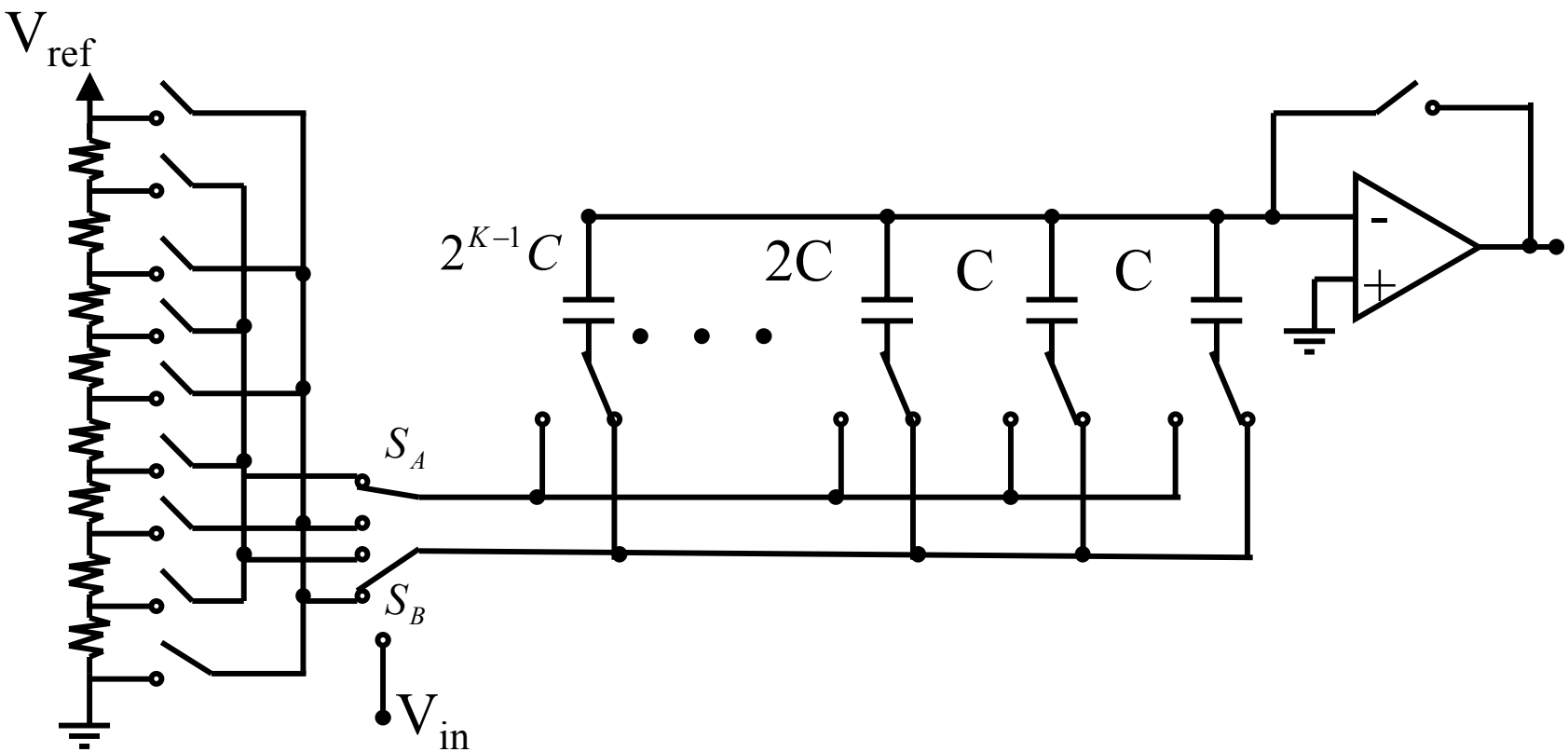
Signed charge-Redistribution ADC with a Single Reference Voltage (cont.)



This approach for realizing signed A/Ds has the disadvantage that V_{in} has been attenuated by a factor of two, which makes noise more of a problem for high-resolution A/Ds. Also, any error in the MSB capacitor now causes both an offset and a sign-dependent gain error. The latter causes integral nonlinearity errors.

Resistor-Capacitor Hybrid ADC

- Combination of resistor-string and capacitor array
- Operation
  1. Charge all the capacitor to V_{in} while the comparator is reset



Resistor-Capacitor Hybrid ADC (cont.)

2. Successive-approximation conversion is performed to find the two adjacent resistor nodes that have voltages larger and smaller than V_{in}. One bus will be connected to one node while the other is connected to the other node.

3. A SA using the capacitor-array network is then done.

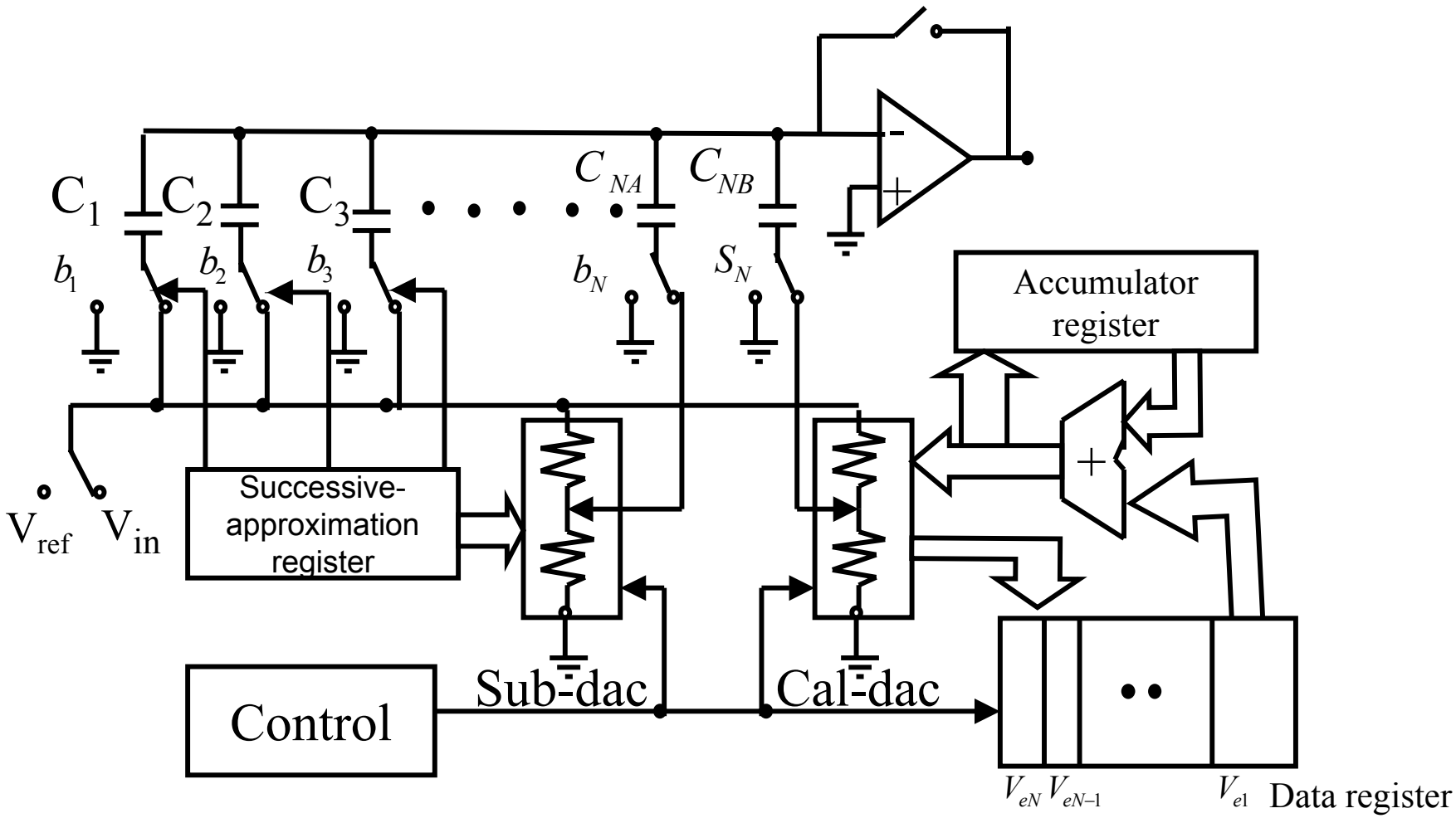
Charge-Redistribution with Error Correction

- Best component matching accuracy is about 0.1%
  - SA converter without calibration can have up to 10-bit accuracy.
  - SA converter with error-correction techniques can have up to 16-bit accuracy.
- Example :16-bit
  - 10 bit MSBs using binary-weighted capacitors.
  - 6 bit LSBs (referred to as sub-dac) using
    1. An additional capacitor and
    2. A resistor string

    No correction terms are measured for the resistor sub-dac; It's accuracy is not critical since it only determined the LSBs.

Charge-Redistribution with Error Correction (cont.)

– The MSB capacitor array is not inherently monotonic but can be easily auto calibrated at start-up by adding a second resistor string (referred to as cal-dac)

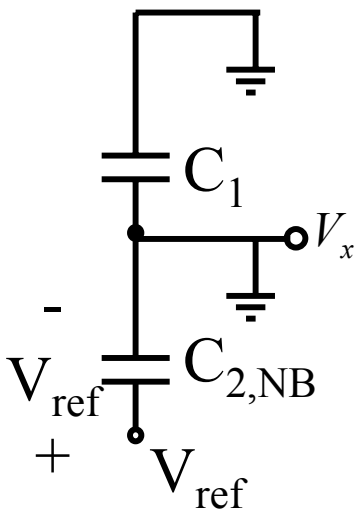


Charge-Redistribution with Error Correction (cont.)

- Calibration
  - calculating the correction terms required, and then storing in a data register as $D_{Vei.}$
  - During a regular SA operation, whenever a particular capacitor is used, its error is cancelled by adding the value stored in the data register to that stored in an accumulator register, which contains the sum of the correction terms for all of the other capacitors currently connected to V_{ref}.

Charge-Redistribution with Error Correction (cont.)

- Calculation of capacitor error
  - starting with MSB capacitor, C_1

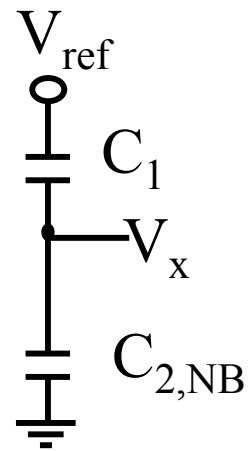


1. Comparator is reset $C_1 \rightarrow$ GND
   all other capacitors $\rightarrow V_{ref}$

$$\Rightarrow \begin{cases} V_x = 0 \\ Q(C_{2,NB}) = C_{2,NB}\, V_{ref} \end{cases}$$

where $C_{2,NB}$=sum of all other capacitors

2. comparator is taken out of reset $C_1 \rightarrow V_{ref}$ all other capacitors $\rightarrow$ GND



$$\Rightarrow C_1(V_{ref} - V_x) + C_{2,NB}V_x = -C_{2,NB}V_{ref}$$

$$\Rightarrow V_x = -\frac{C_1 - C_{2,NB}}{C_1 + C_{2,NB}} Vref$$

Charge-Redistribution with Error Correction (cont.)

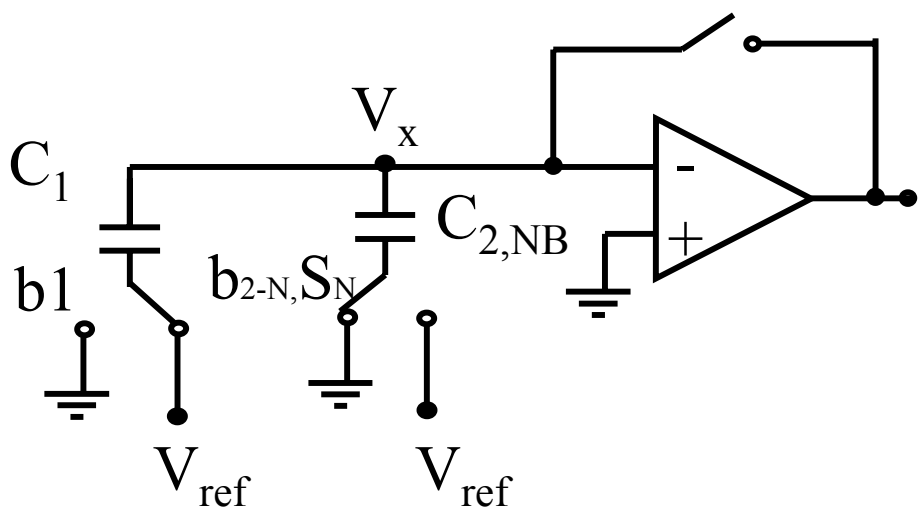
3.Define C_{total} =sum of all capacitors

$$C_1 = \frac{C_{total}}{2} + \Delta C_1, \quad C_{2,NB} = \frac{C_{total}}{2} - \Delta C_1$$

$$\Rightarrow DV_x = V_x = \frac{2\Delta C_1}{C_{total}} V_{ref} = \underbrace{2\left(\frac{C_1}{C_{total}} - \frac{1}{2}\right)V_{ref}}_{V_{e1}}$$

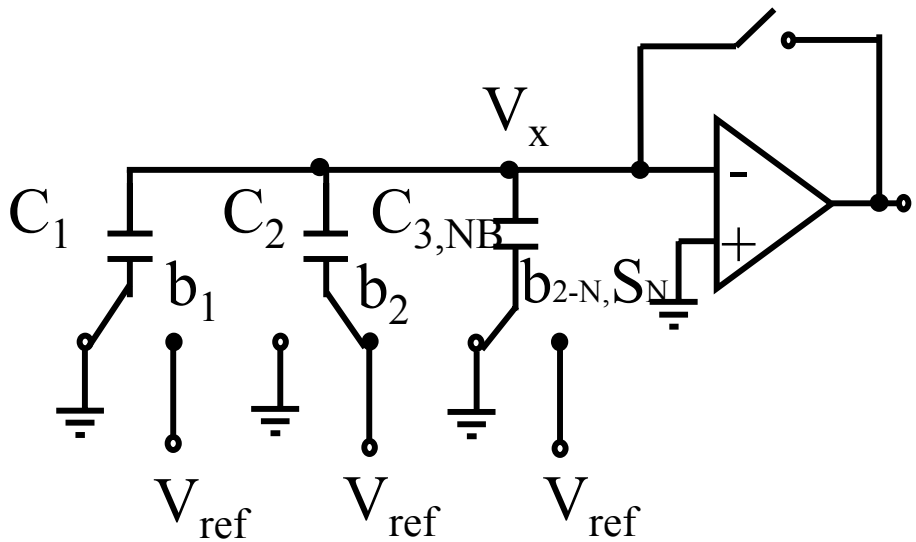
$$\Rightarrow DV_x = 2V_{e1}$$

{ The resulting voltage V_x is twice the error voltage V_{e1} }



Charge-Redistribution with Error Correction (cont.)

- "digital representation of V_{e1}"=DV_{e1} is obtained by doing a successive approximation using the cal-dac and then dividing the resulting digital value by 2 to obtain DV_{e1}.
- DV_{e1} is then stored in the data register for use during regular conversion.
  - Correction term of C_2
    1. comparator is reset
       $C_1 \rightarrow$ GND
       $C_2 \rightarrow$ GND
       all others $\rightarrow V_{ref}$



$$C_2 = \left(\frac{C_{total}}{4}\right) + \Delta C_2$$

$$C_{3,NB} = \left(\frac{C_{total}}{4}\right) - \Delta C_1 - \Delta C_2$$

Charge-Redistribution with Error Correction (cont.)

2. comparator is taken out of reset

   $C_1 \rightarrow$ GND

   $C_2 \rightarrow V_{ref}$

   all other capacitor $\rightarrow V_{ref}$

3. $DV_{e2}=0.5(DV_{x2}-DV_{e1})$ where DV_{x2} is the measured V_{x2}

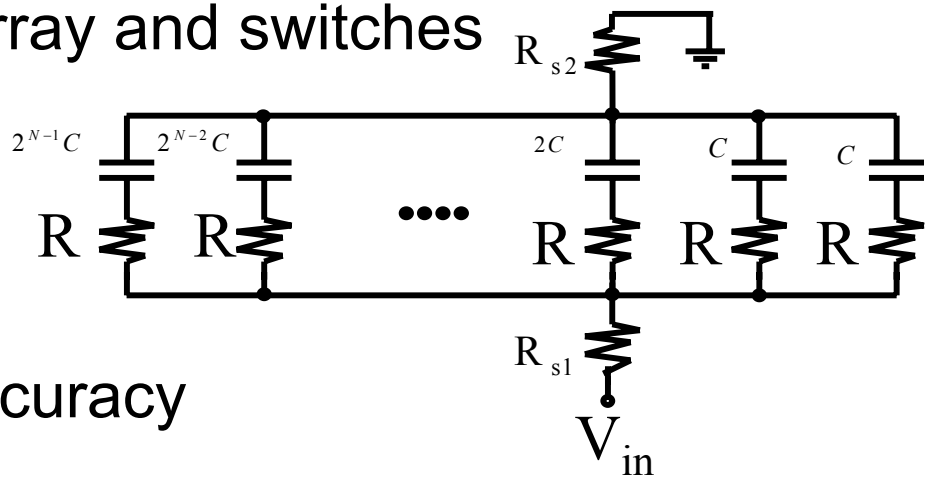
– formula

$$DV_{ei} = 1/2(DV_{xi} - \sum_{j=1}^{i-1} DV_{ej})$$

Speed Estimate for Charge-Redistribution ADC

- The major limitation on speed is due to the RC time constants of the capacitor array and switches
- Simplified model
- open-circuit time constant

$$\tau_{eq} = (R_{s1} + R + R_{s2})2^N C$$



– for better than 0.5LSB accuracy

$$e^{-T/\tau_{eq}} < \frac{1}{2^{N+1}}$$

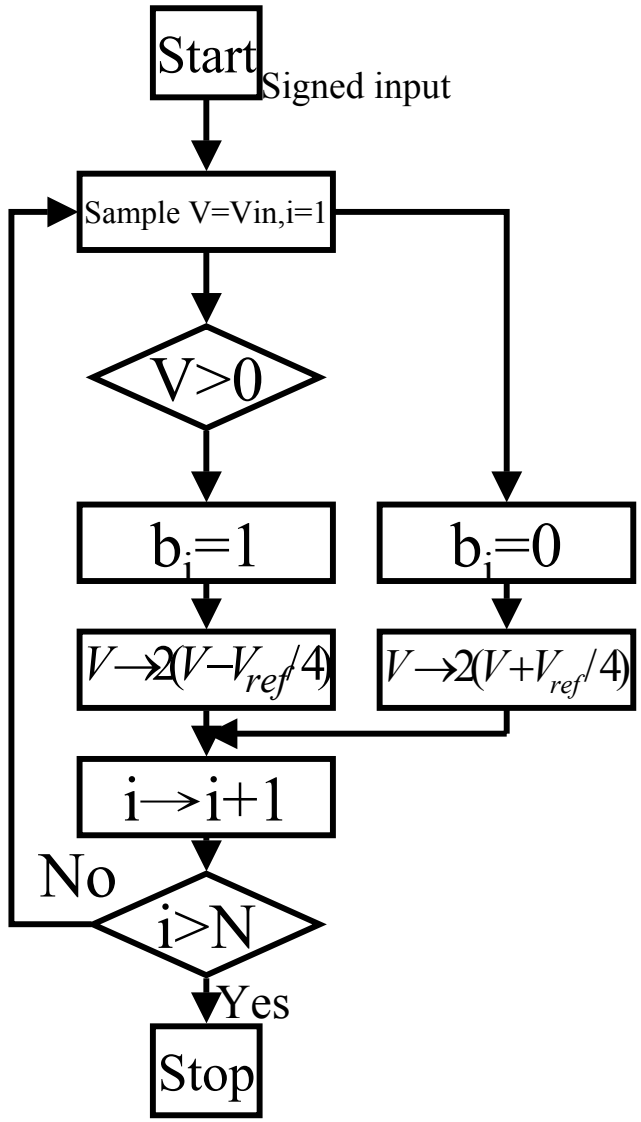
$$T > \tau_{eq}(N+1)\ln(2) = 0.69(N+1)\tau_{eq}$$

→30% higher than actual value

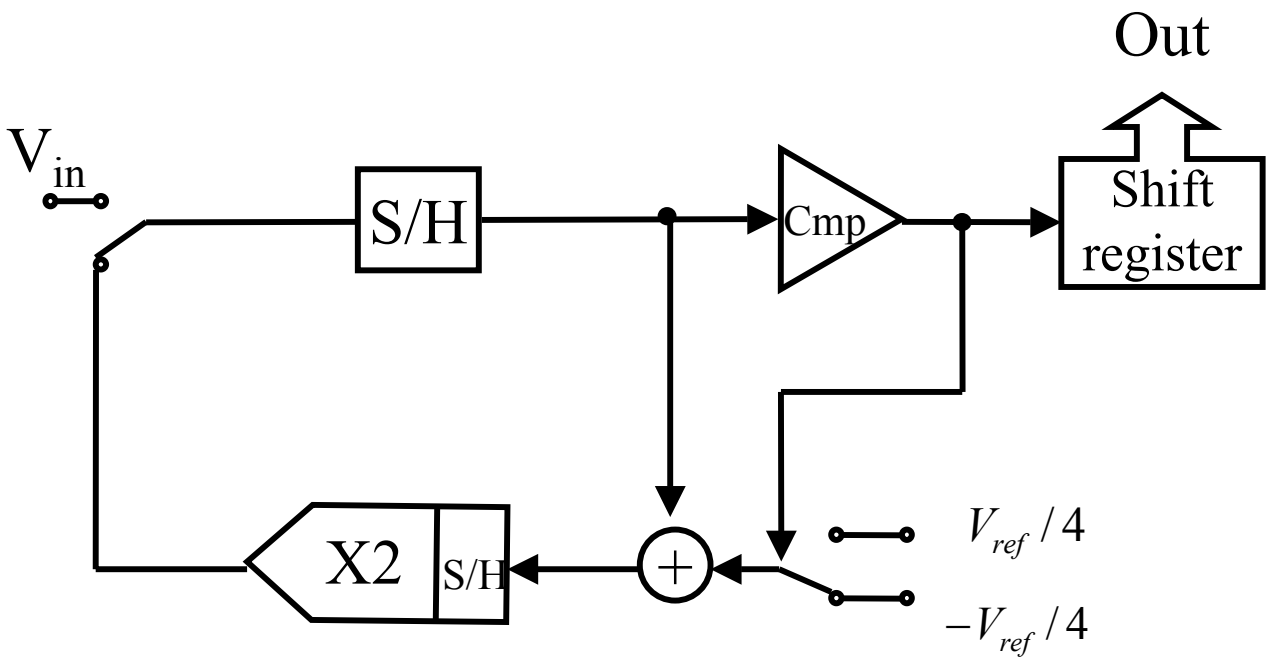
→Circuit simulation for the ADC is required to obtain real speed

Algorithmic(or cyclic)ADCs

- A successive approximation converter halves the reference voltage in each cycle, an algorithm converter doubles the error voltage while leaving the reference voltage uncharged
- Flow graph
- Block diagram
  - requires a small amount of analog circuitry because it repeatedly used the same circuitry to perform its conversion cyclically in time
  - accurate multiply-by-two gain amplifier
    1. four clock cycles are required.
    2. does not rely on any capacitor matching



Algorithmic(or cyclic)ADCs (cont.)

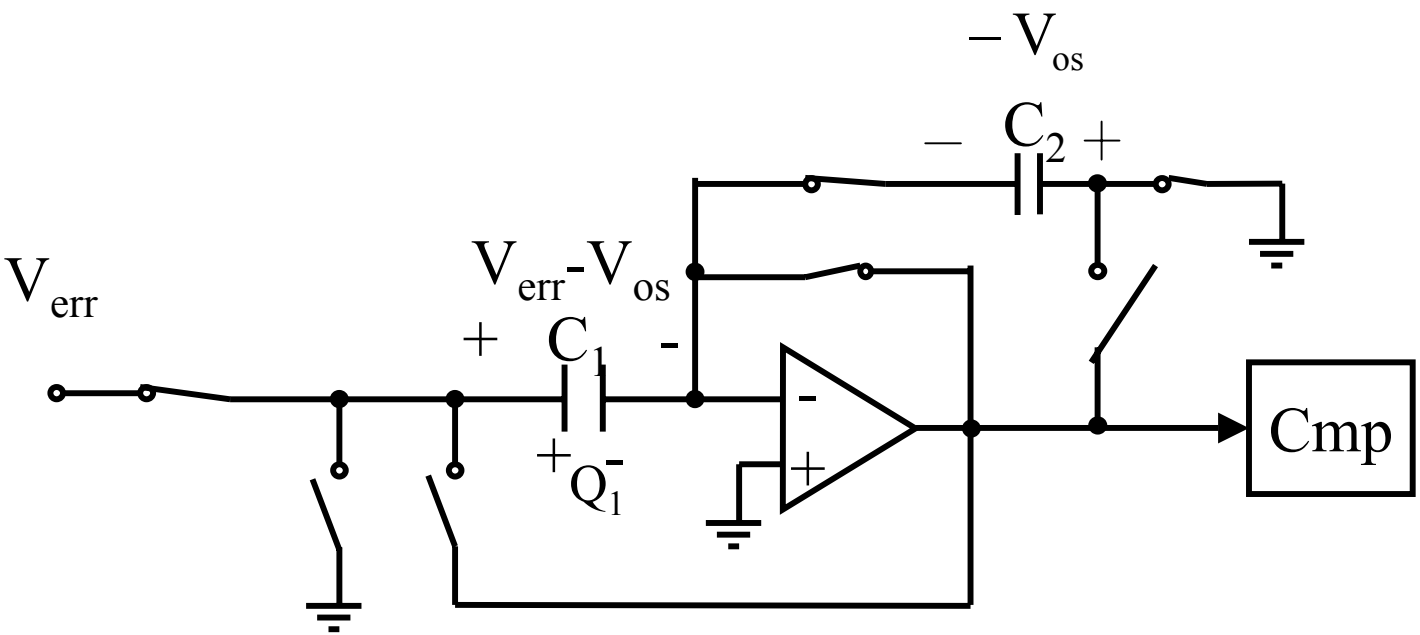


Multiply-By-Two Gain Circuitry

- Fully-differential circuits are normally used

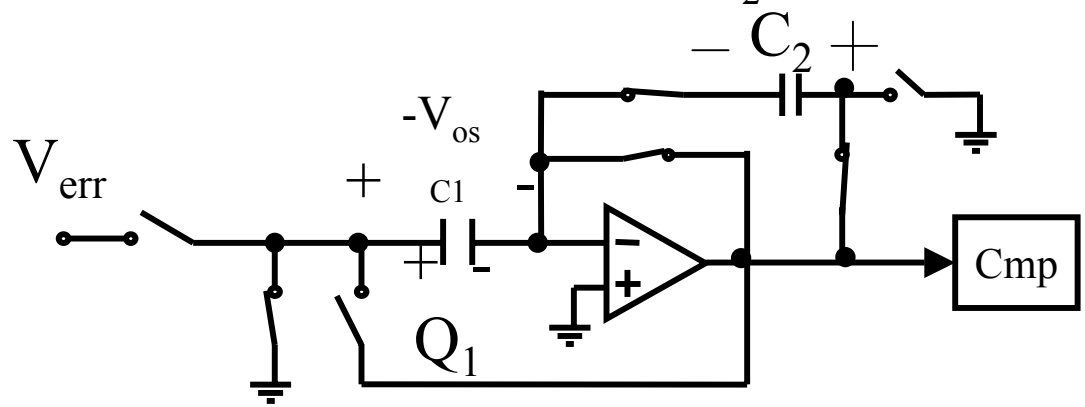
  (For simplicity, a single-ended circuit is used here)
- Operational principle (four phases)

  1. sample remainder and cancel input-offset voltage

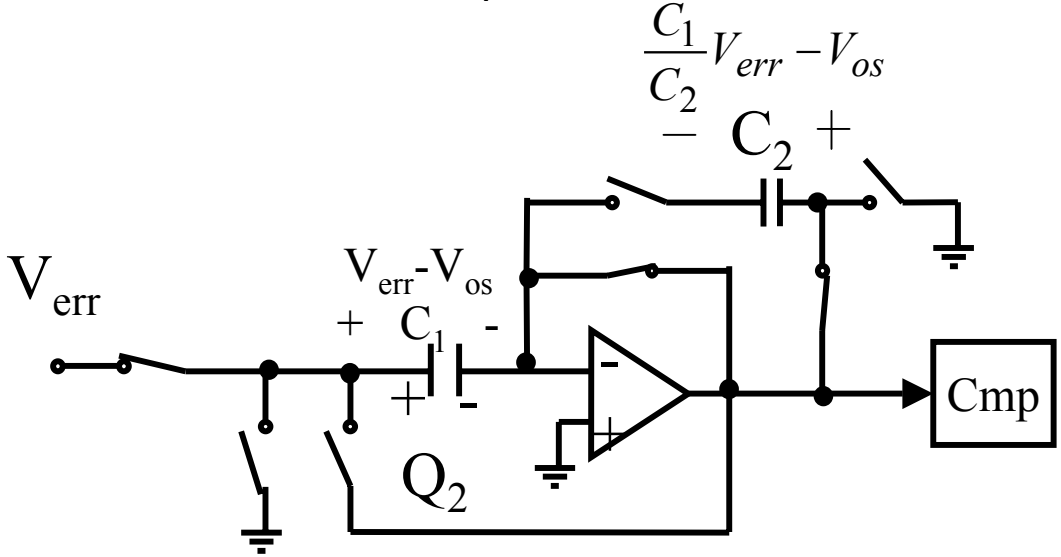


Multiply-By-Two Gain Circuitry (cont.)

2. Transfer charge Q_1 from C_1 to C_2 $\frac{C_1}{C_2}V_{err} - V_{os}$

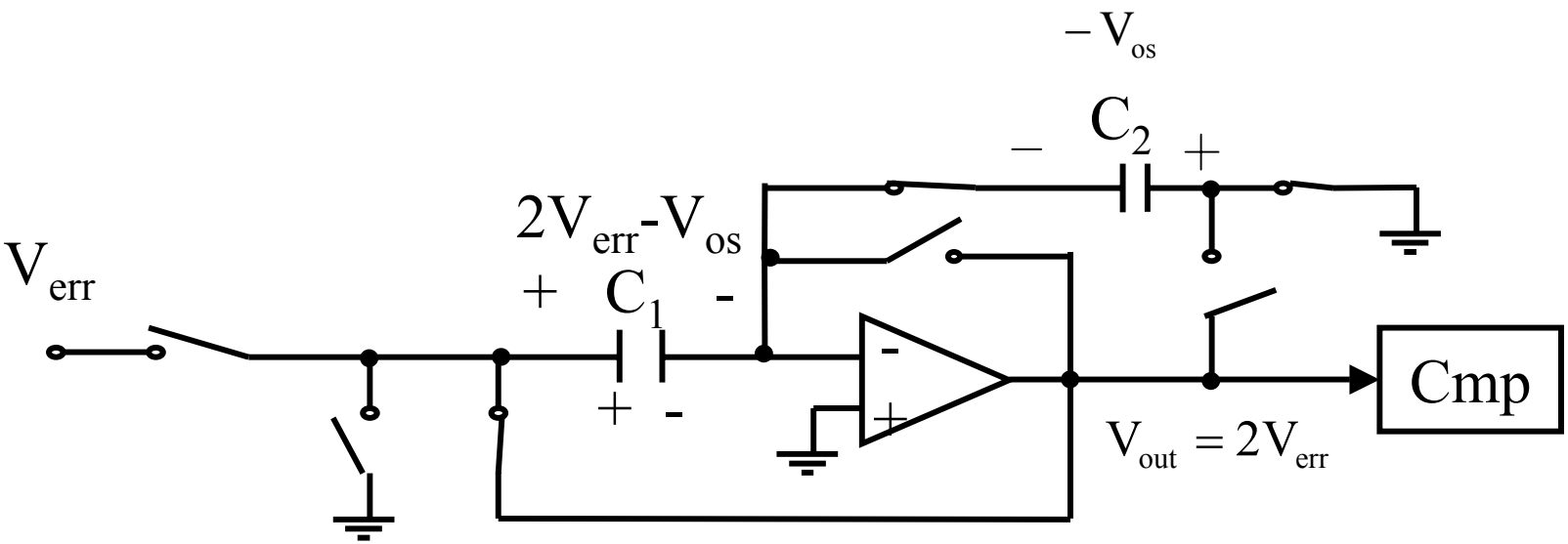


3. sample input signal with C_1 again, after storing Q_1 on C_2



Multiply-By-Two Gain Circuitry (cont.)

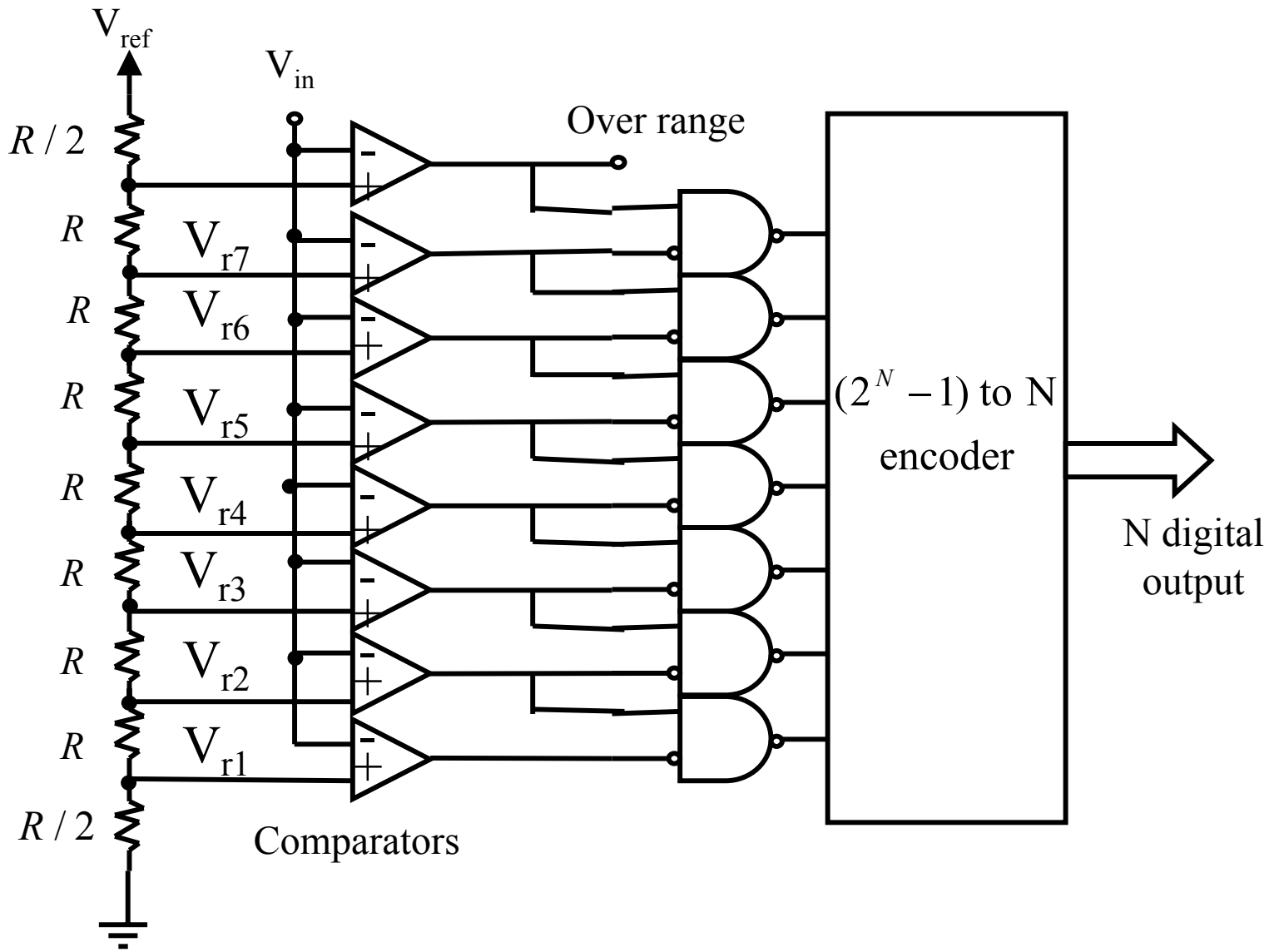
4.Combine Q_1 and Q_2 on C_1, and connect C_1 to output



Flash(or parallel)ADC

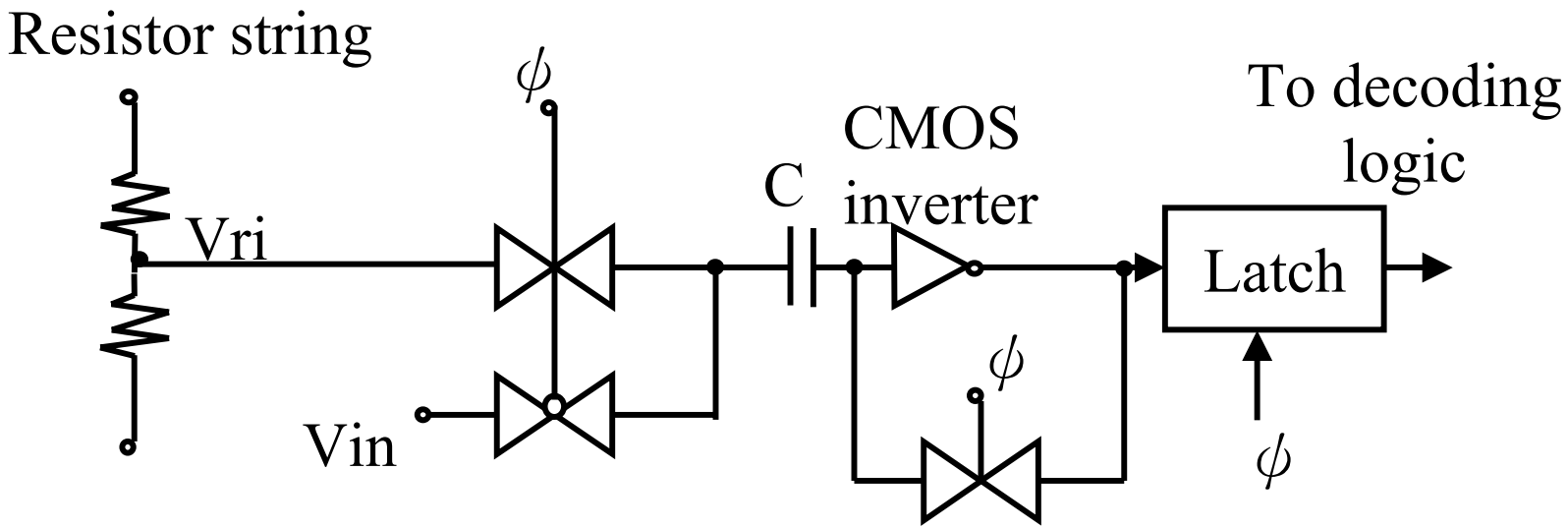
- Very-high-speed approach, especially in 1980s.
- Large area and power hungry.
  - 2^N comparators
  - 2^N reference voltages, V_{r1}, V_{r2}, ….,generated by a resistor string
- Thermometer code at comparator outputs
  - 2^N-1 NAND gates to detect the transition of the comparator output from 1s to 0s.
    1. The NAND gate that detects a transition will have a 0 output.
    2. all other NAND-gate output will be 1
  - bubble error occurs if more than one 0 output is obtained

Flash(or parallel)ADC (cont.)



Flash(or parallel)ADC (cont.)

- CMOS example using clocked comparator
  - its inverts as a single stage OPAMP with only one pole

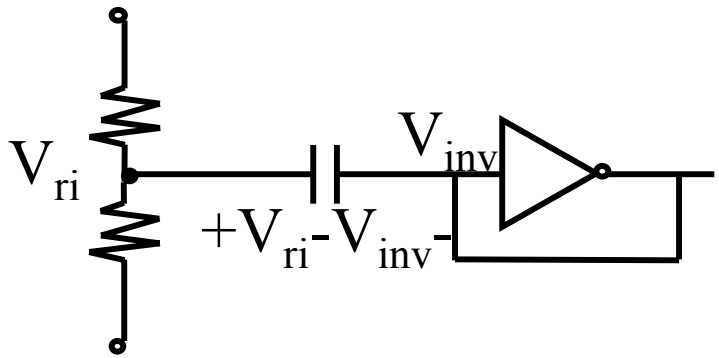


Flash(or parallel)ADC (cont.)

– Two operation phases

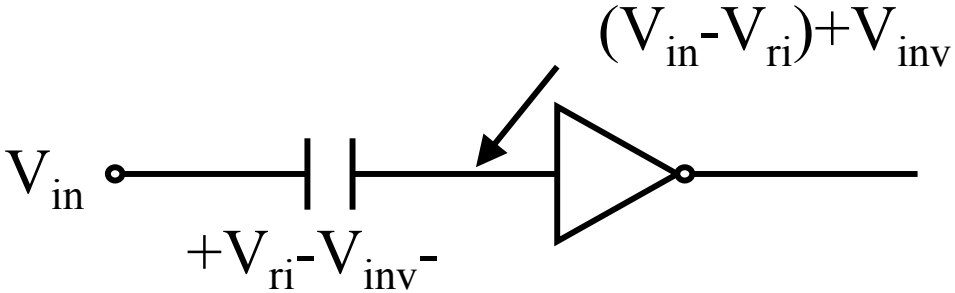
1.autozero (ϕ =1)

with the inverter set to its threshold, V_{inv},the other side of C is charge to V_{ri}



2.Signal sampling & conversion (ϕ =0)

(V_{in}-V_{ri})determines which direction the inverter output will fall



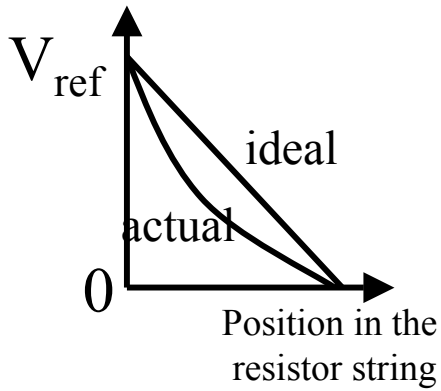
Flash(or parallel)ADC (cont.)

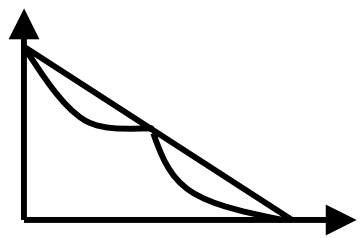
- This simple comparator suffers from poor power supply rejection. Full differential inverter helps alleviate this shortcoming.
- The inverter gain must be large enough to amplify (V_{in}-V_{ri}) to V_{iH} and V_{iL} of its succeeding latches. Usually, gain $=25\sim100$ for 8-bit resolution. Most often, 2 cascade inverters are used to optimal speed. Each has a gain of 5~10. Inverter are autozeroed individually.

Issues in Designing Flash ADC

- Large input capacitive load
  - large number of comparators connected to Vin
  - often limit the speed
  - usually requires a strong and power hungry buffer to drive Vin
  - can be used by using other structures, e.g. two-step, interpolating, pipeline,….etc.
- Resistor string bowing
  - input currents of bipolar comparator

    currents required to charge C during autozreo phase of clocked COMS comparators
  - errors are greatest at the center node of the resistor string
  - considerable improvement obtained forcing the center tap voltage to be correct. However, more voltage references are required.





Issues in Designing Flash ADC (cont.)

- Comparator Latch-to-Track delay
  - especially when a small input signal of the opposite polarity from the previous period is present
  - can be minimized by keeping the time constants of the internal nodes of the latch as small as possible. This is sometimes achieved by keeping the gain of the latches small, e.g. 2~4
  - differential internal nodes might be stored together temporarily just after latch time.

Issues in Designing Flash ADC(Cont.)

- Signal and/or clock delay
  - Even very small differences in the arrival of clock or input signals at the different comparators can cause errors.

    e.g. An 8-bit ADC with V_{ref}=2V.

    For a 250MHz 1V peak-to-peak input(sinusoid), it takes 5ps to change 1 LSB which is about the same time for a signal to propagate 500 μm in metal interconnect .
    If there is clock skew between comparators greater than this, the converter will have more than 1 LSB error.

Issues in Designing Flash ADC(Cont.)

– To reduce this error

1. Using S/H
   However, high-speed S/H can be more difficult to realize than the flash converter itself.
2. The clock and V_{in} should be routed together with the delay matched . However, delay differences could also be caused by different capacitive loads, or by phase differences between the comparator preamplifiers at high frequencies.

Issues in Designing Flash ADC(Cont.)

- Substrate and power supply noise
  - 7.8mV of noise injection would cause a 1LSB error for an 8-bit convertor with V_{ref}=2V.

    On an IC having a clock signal in the tens of MHz, it is difficult to keep power-supply noise below a few tens of a volt.
  - To reduce this effect

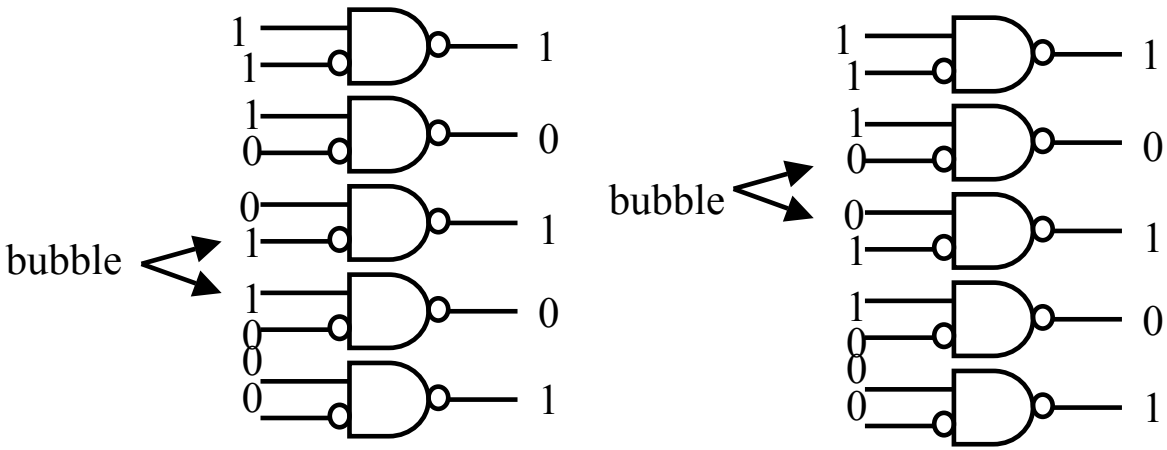
    1. Running differential clocks closely together will help prevent the signals being coupled into the substrate or through the air.

    2. Analog power supplies should be separated from digital power supplies including having analog power to the comparator preamps while using digital power to the latch stages.

Issues in Designing Flash ADC(Cont.)

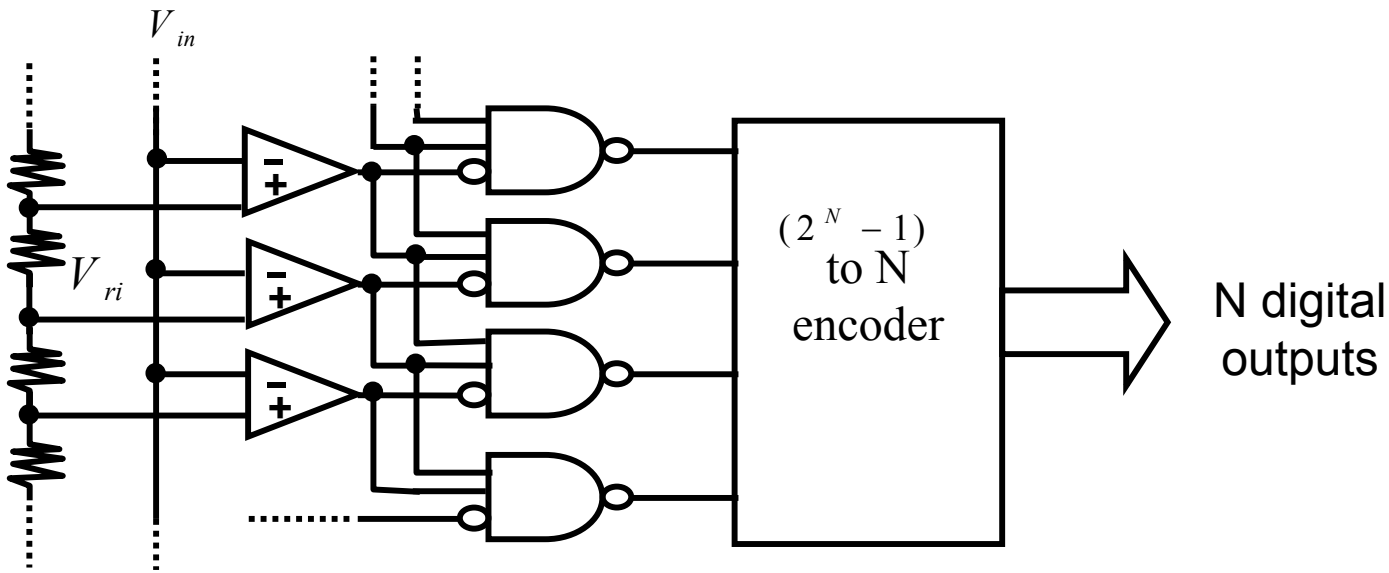
3 .On-chip power-supply bypassing is a necessity
make sure the power-supply bypassing circuitry doesn't form a resonant circuit with the bonding wire.

- Bubble error removal
  - error due to comparator metastability, noise, cross talk, limited bandwidth, …,etc.
  - bubble examples



Issues in Designing Flash ADC(Cont.)

– Can be removed using 3-bit NAND gates if bubbles occur near the transition point of the thermometer code.

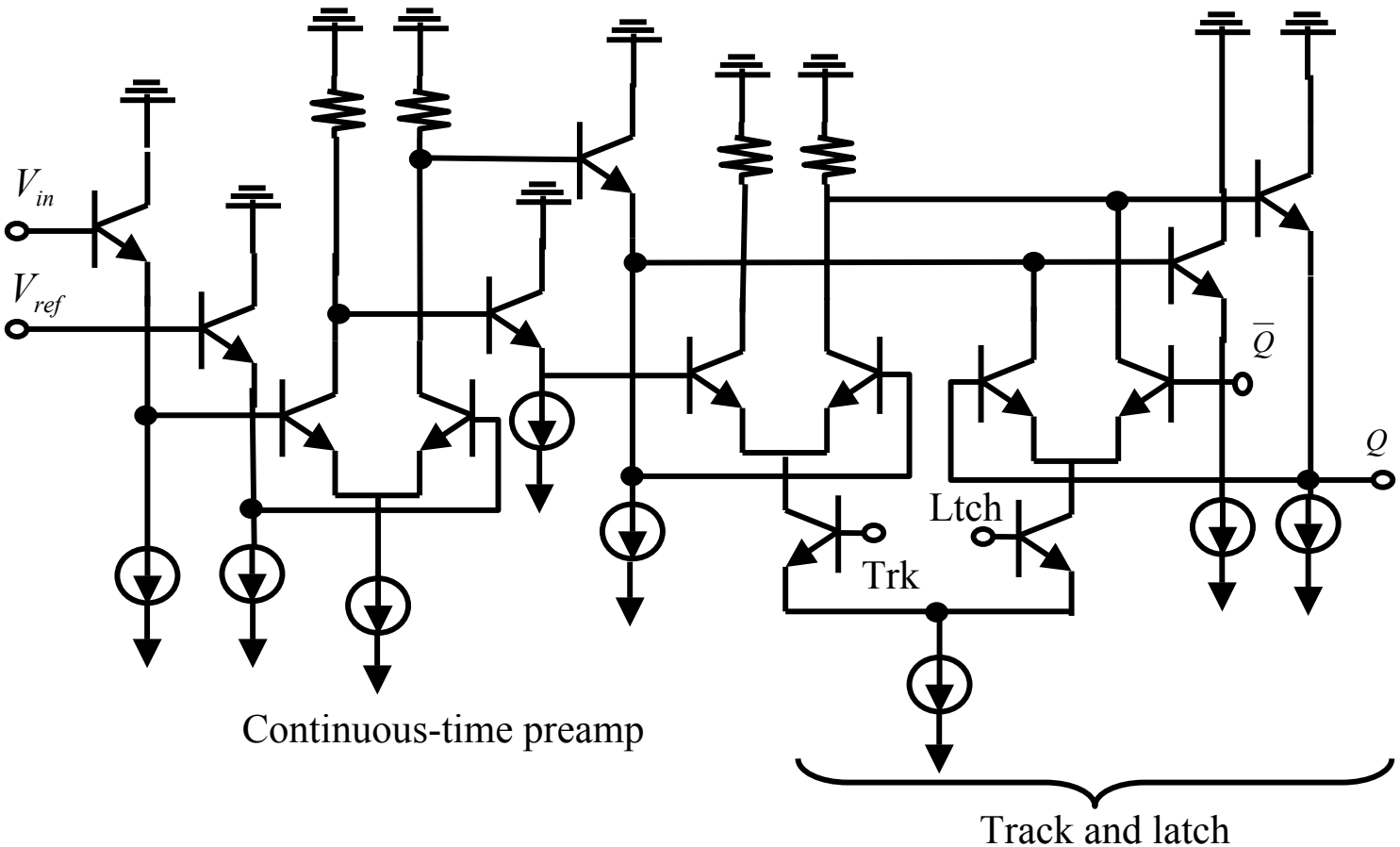


– Distant bubble errors can also be reduced using other approaches in p.511 of textbook.

Issues in Designing Flash ADC(Cont.)

- Flashback
  - caused by latched comparators when they are switched from track to latch mode.
  - charge glitch at the inputs to the latch.
  - If there is no preamplifier, this will cause major errors due to the unmatched impedance at the comparator inputs.
  - To minimize this effect, most modern comparators have one or two stages of continuous-time buffering and/or preamplification.

Issues in Designing Flash ADC

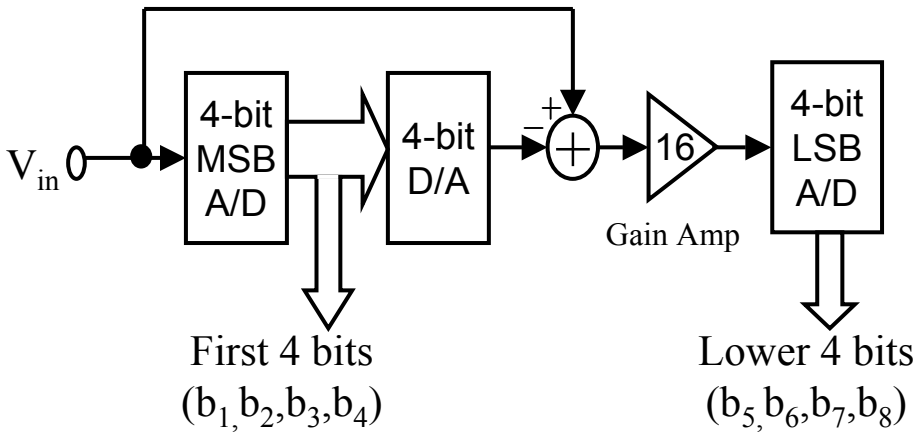


Two-Step (or Subranging) ADC

- Compared to flash ADC
  - currently more popular
  - less area
  - less power
  - less input capacitive loading
  - the voltages the comparators need to resolve are less stringent
  - larger latency
  - can't have very high speed due to the use of S/H

Two-Step (or Subranging) ADC(Cont.)

- 8-bit example



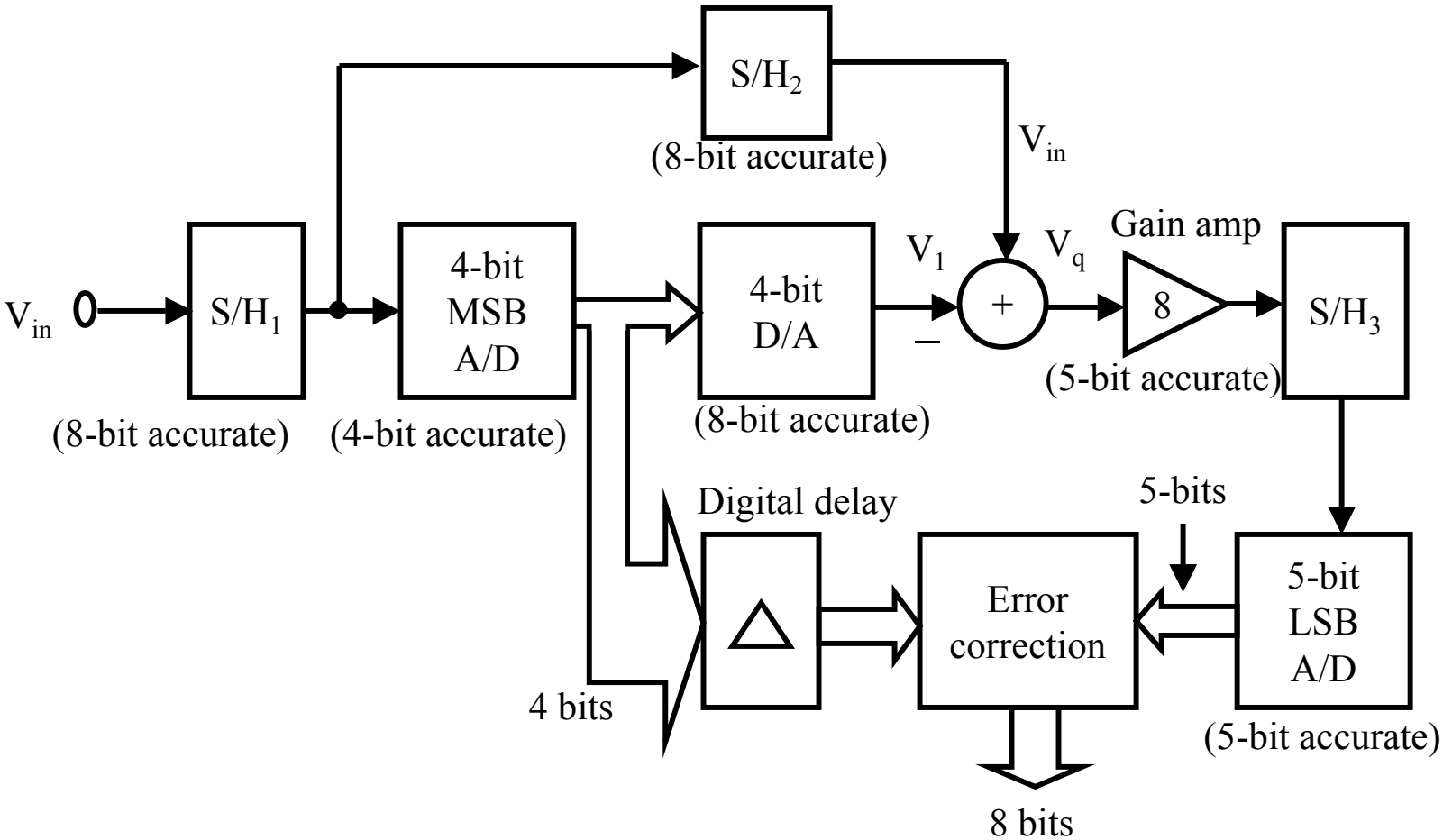
  - the 4-bit MSB A/D determines the first four MSBs
  - to determine the remaining LSBs
    1. the quantization error, V_q, of the MSB A/D is further converted.
    2. V_q is multiplied by 16 to ease circuit requirements for finding LSBs.
    3. the LSBs are determined using the 4-bit LSB A/D

Two-Step (or Subranging) ADC(Cont.)

– This straightforward approach would require all components to be at least 8-bit accurate. To significantly ease the accuracy requirements of the 4-bit MSB A/D, digital error correction is commonly used.

Two-Step (or Subranging) ADC(Cont.)

- Example using digital error correction



Two-Step (or Subranging) ADC(Cont.)

- Signal is pipelined
  1. need more S/H to maintain high speed
  2. speed is halved if only one S/H is used
- with error correction, relaxed circuit accuracy of internal ADCs and gain stage
- accuracy required for each block is shown in the figure above (The reasons are included in p.515 of the textbook)

Interpolating ADC

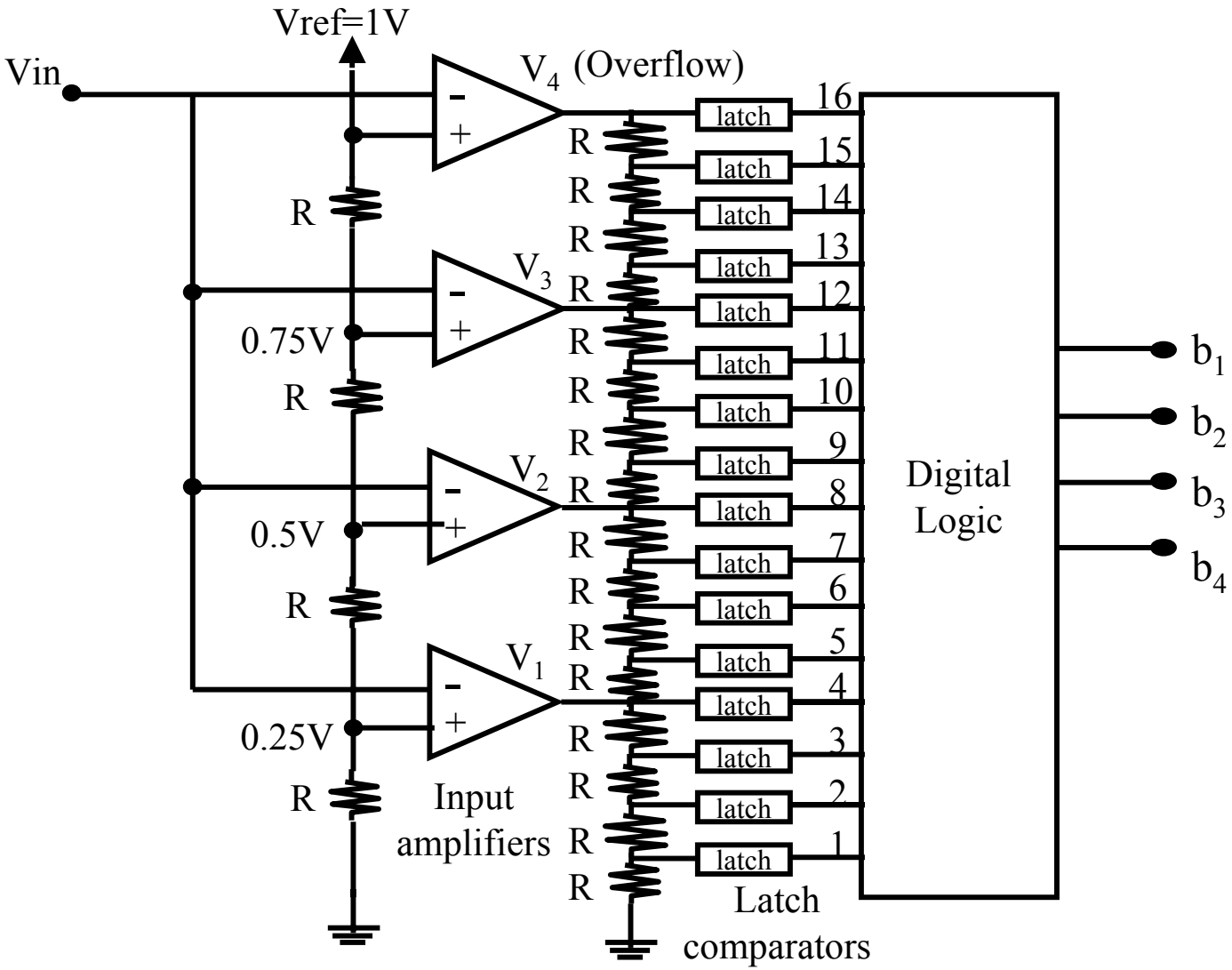
- Compared to flash ADC
  - lower input capacitance
  - slightly reduced power
  - lower number of reference voltages needed
- Use of input amplifiers
  - these amplifiers behave as linear amplifier near their theshold voltages but are allowed to saturate once their differential input become moderately large.
  - the number of input amplifiers attached to V_{in} is significantly reduced by interpolating between adjacent outputs of these amplifiers.

Interpolating ADC(Cont.)

- Example: 4-bit
  - the input amplifiers have a maximum gain of -10

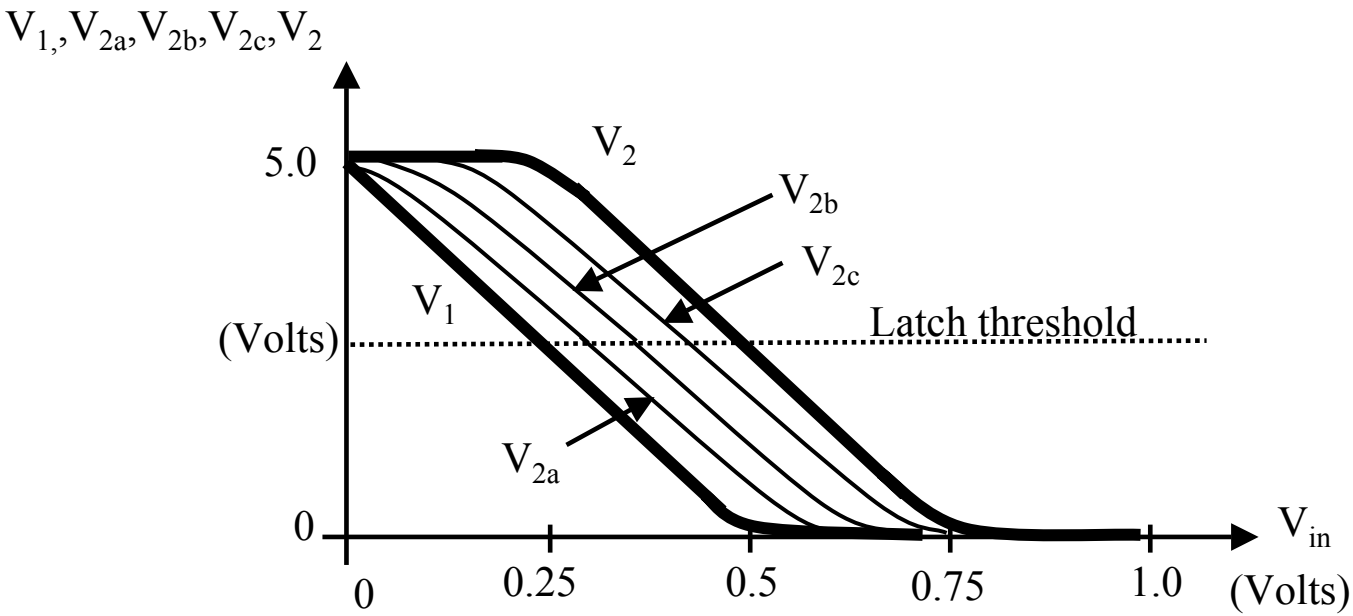
    1. logic level = 0V, 5V
    latch threshold $\approx$ 2.5V
    voltage difference between adjacent nodes of resistor-string = 0.25V
    2. $|gain| \le \frac{2.5V}{0.25V} = 10$
  - For good linearity, the interpolated signals need only cross the latch threshold at correct points, while the rest of the interpolated signals response are of secondary importance.

Interpolating ADC(Cont.)



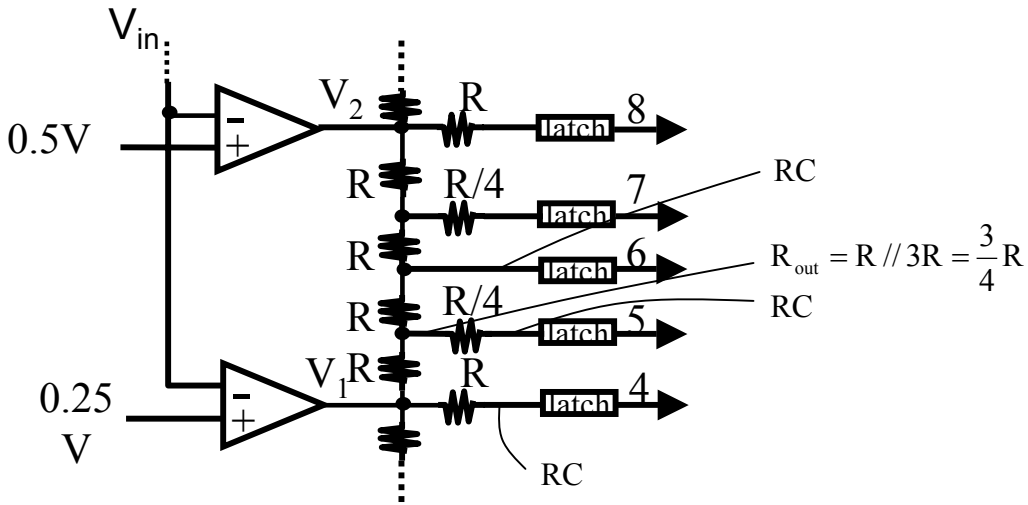
Interpolating ADC(Cont.)

⟹ Linear region corresponds to 0.25V < V_{in} < 0.5V for the bottom linear amplifier



Interpolating ADC(Cont.)

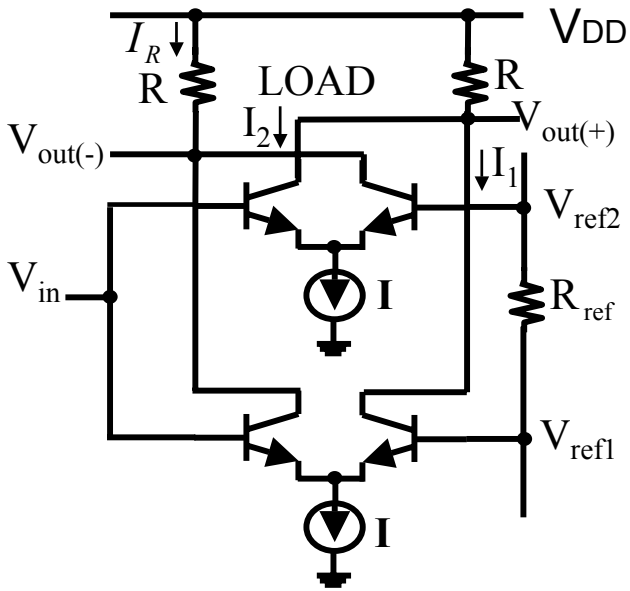
- Delay times equalization
  - delays can be made nearly equal by adding extra series resistors such that the impedances seen by each latch looking into the resistor string, assuming the input-amplifier outputs are low impedance.

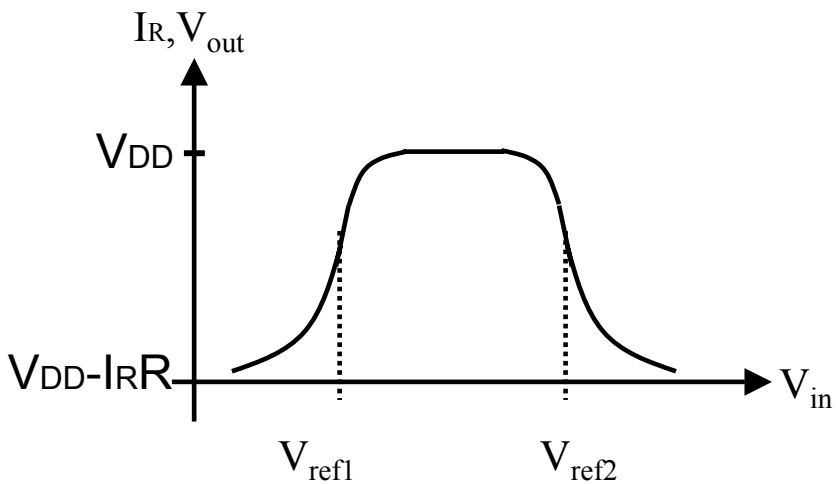


- Other implementation methods
  - interpolating using current mirrors or capacitors

Folding ADC

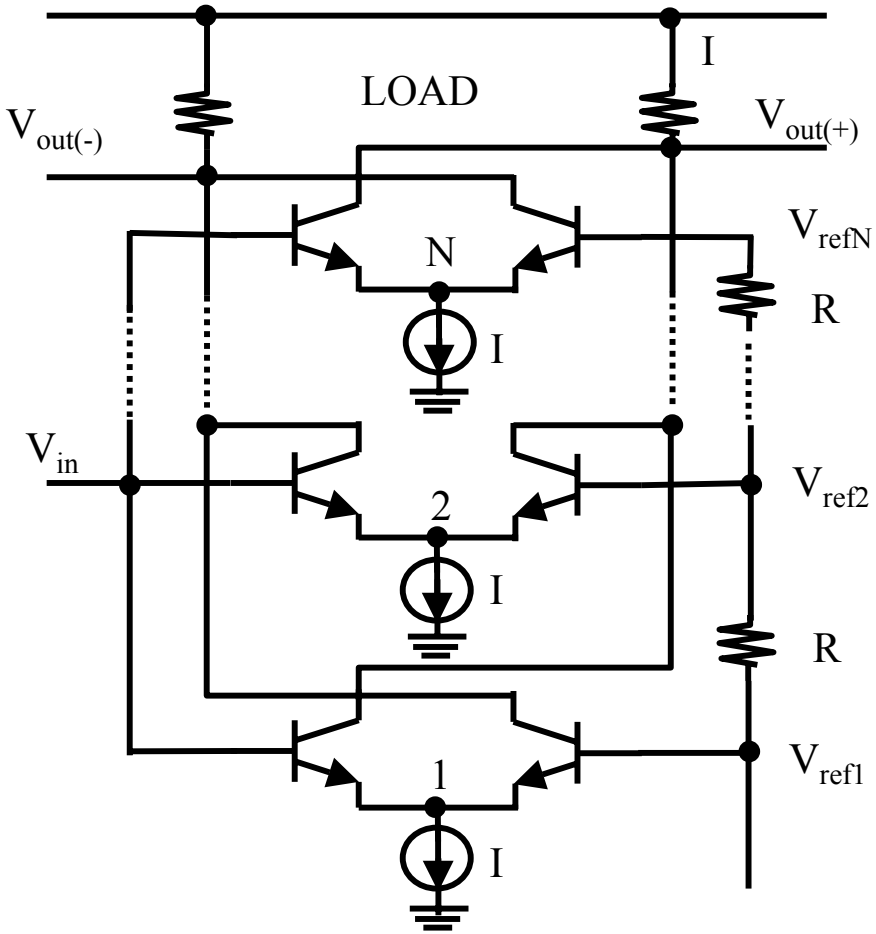
- Heavy input load (similar to flash and heavier than interpolating)
- Reduced number of latch comparators(compared to flash and interpolating)
- Example1
  - two-folded curve generation

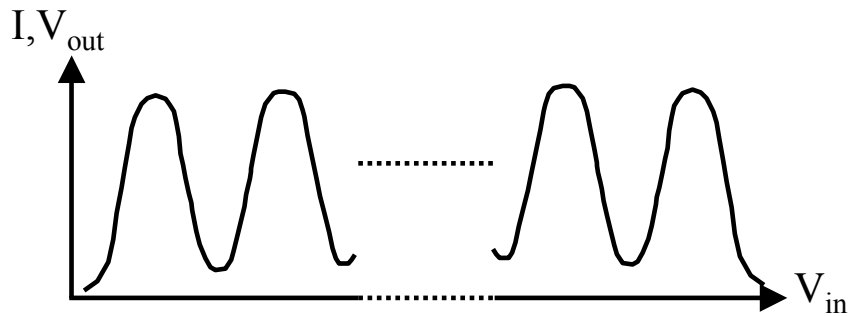




Folding ADC(Cont.)

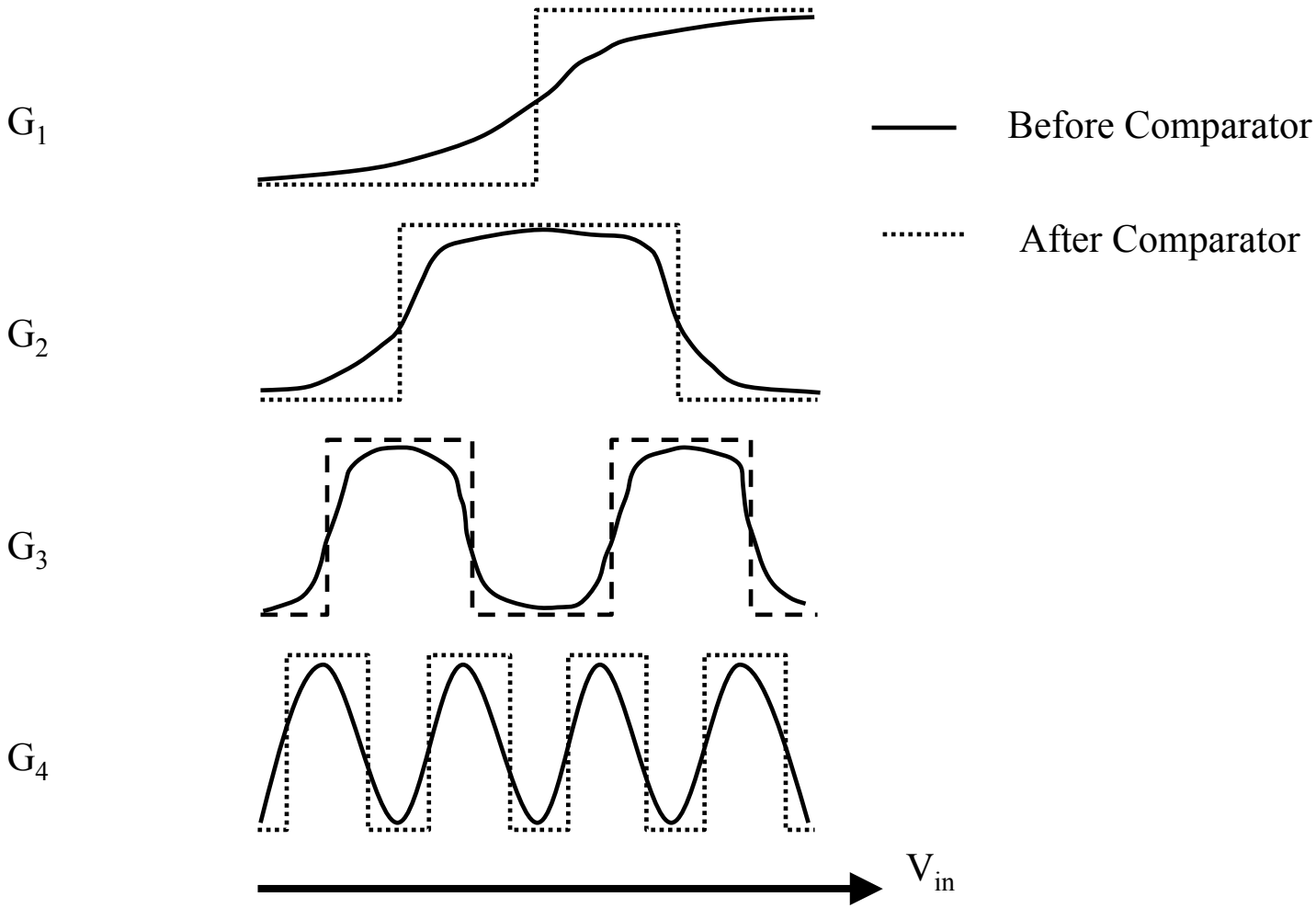
– Multi-folded curve generation





Folding ADC

– Gray coded curves



Folding ADC

– Gray to binary converter
The relation between a Gray code and a binary code

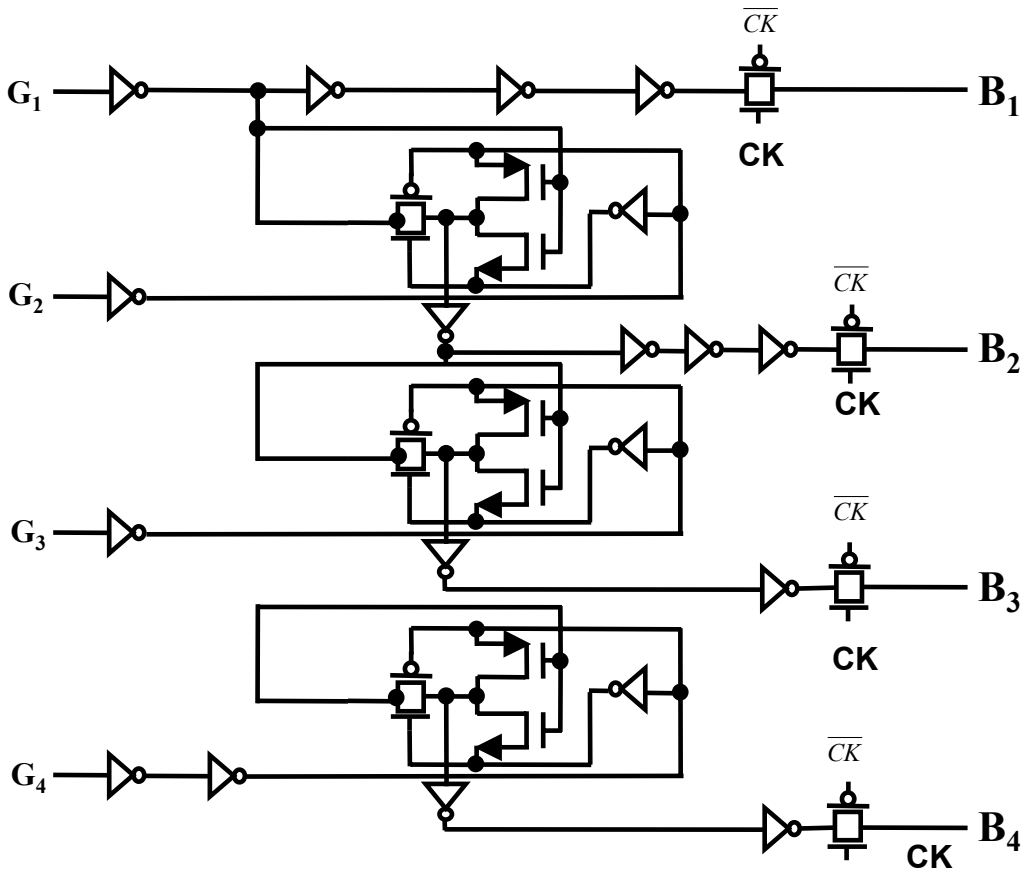
$B_1=G_1$

$B_2=G_2 \otimes B_1$

.....

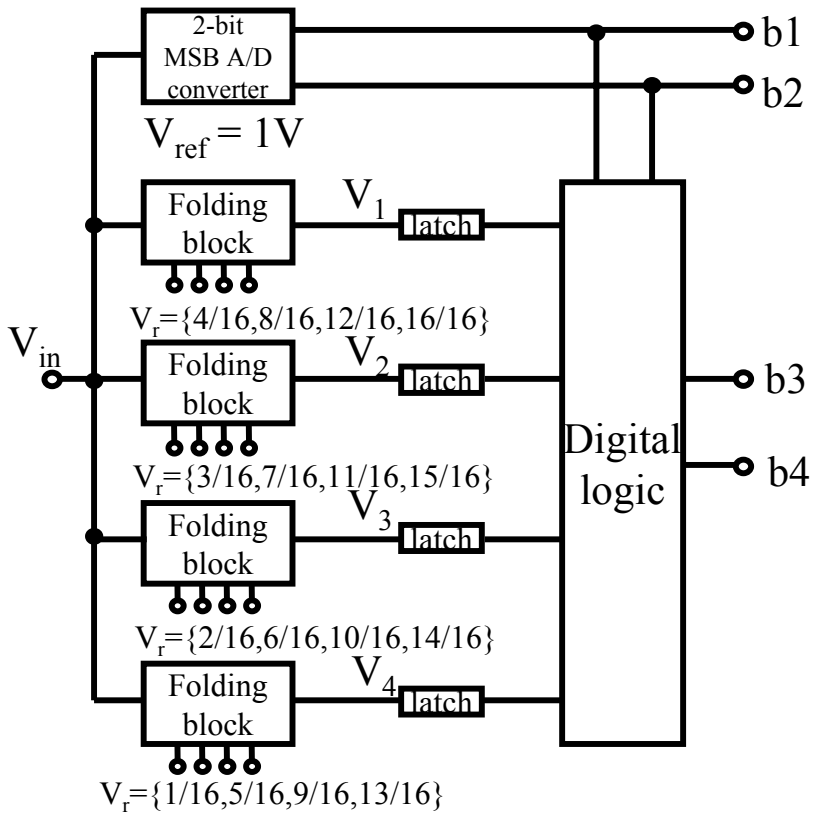
$B_n=G_n \otimes B_{n-1}$

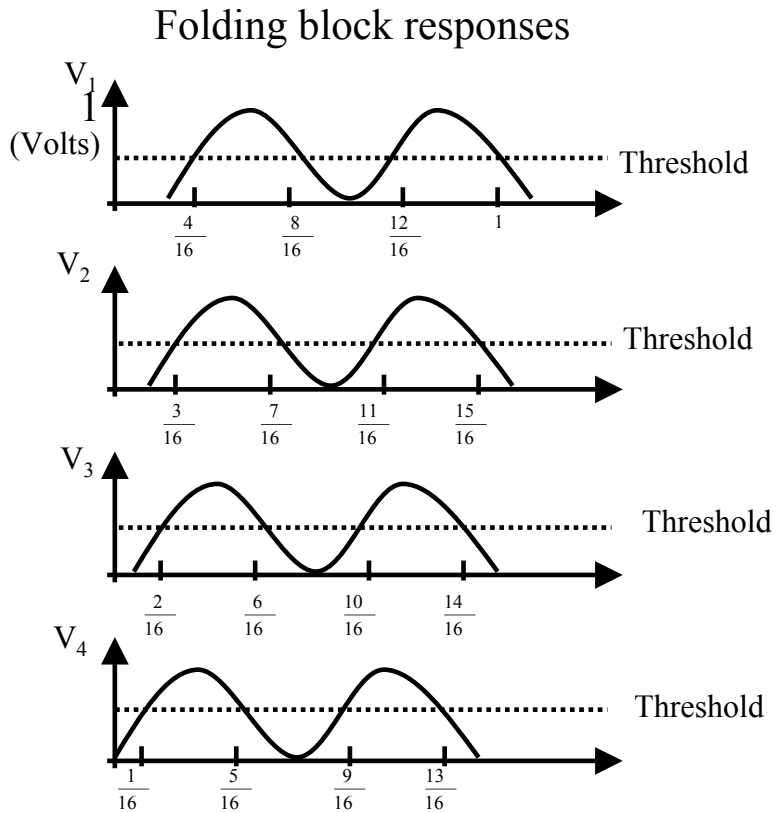
where G_n is Gray bit and
B_n is binary bit
$\otimes$ is exclusive or



Folding ADC(Cont.)

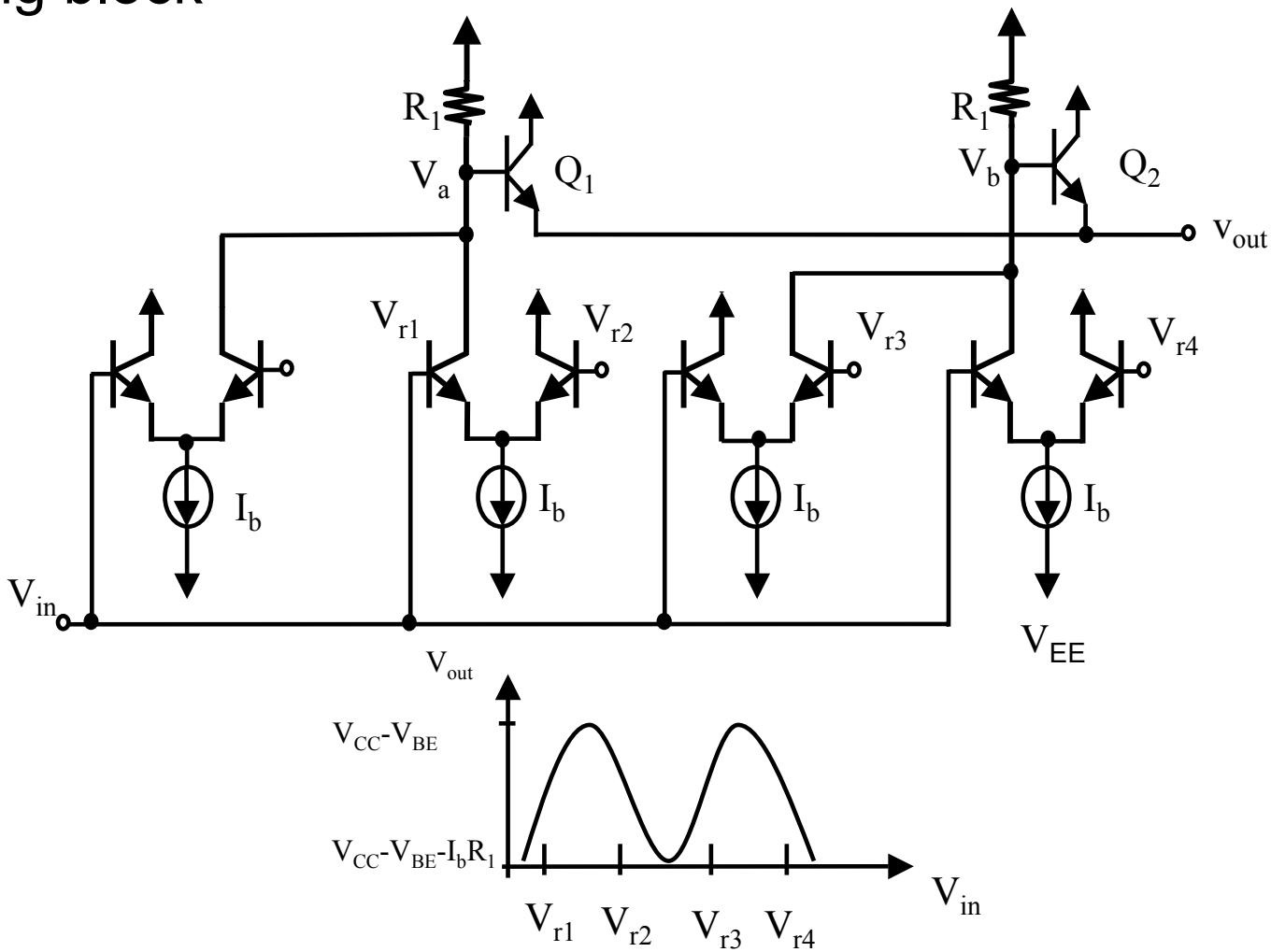
- Example2: A 4-bit ADC with a folding rate of four.
  - The MSB converter would usually be realized by combining some folding block signals, such as V_1 is used to determine $2MSB_s$ in this example.





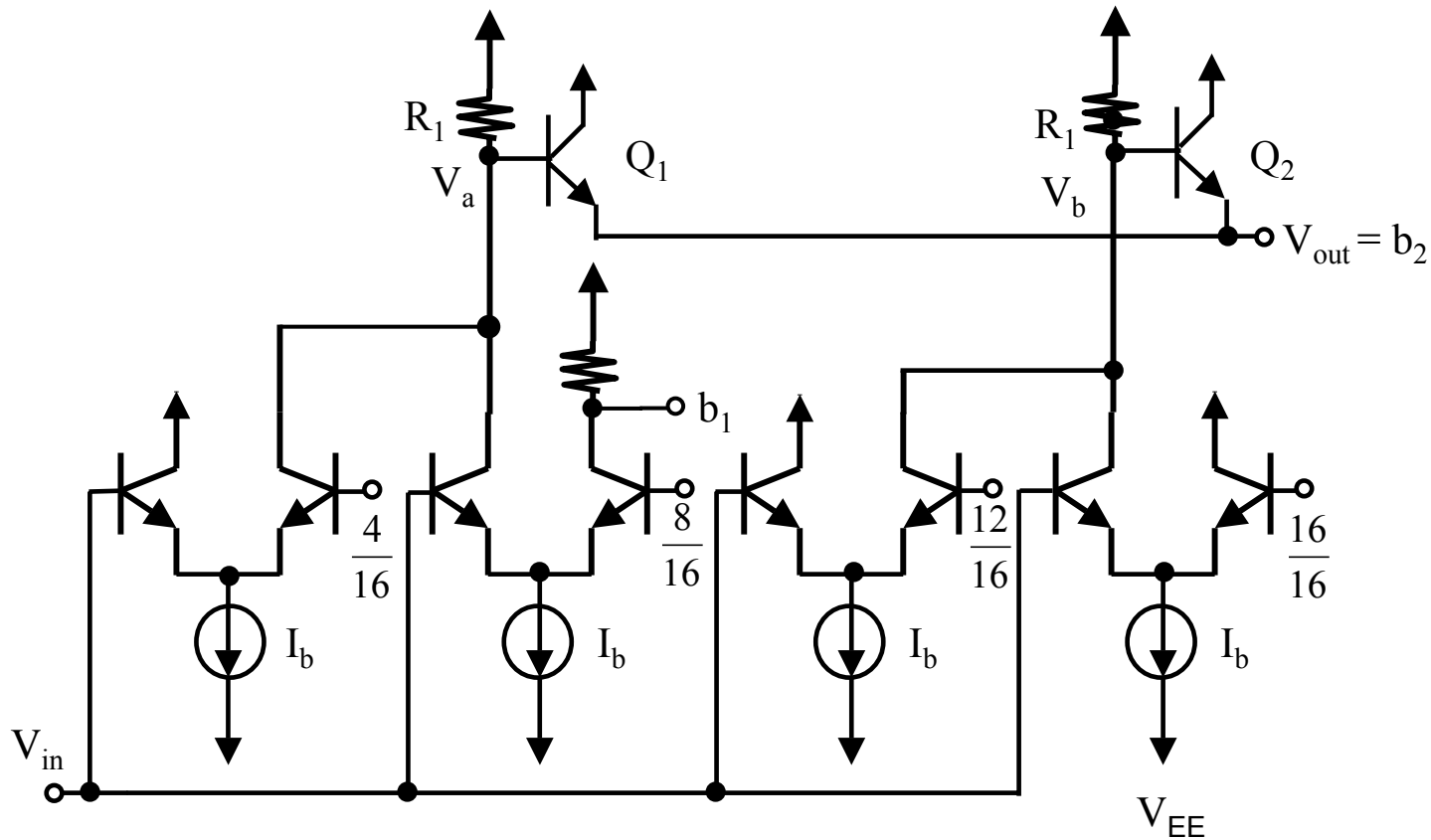
Folding ADC

– Folding block



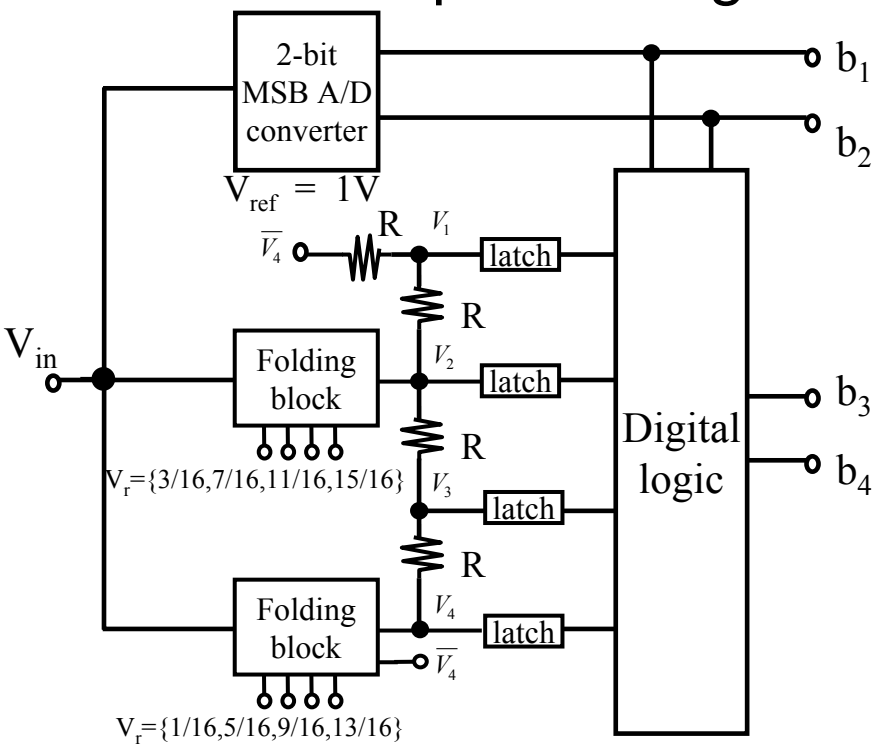
Folding ADC(Cont.)

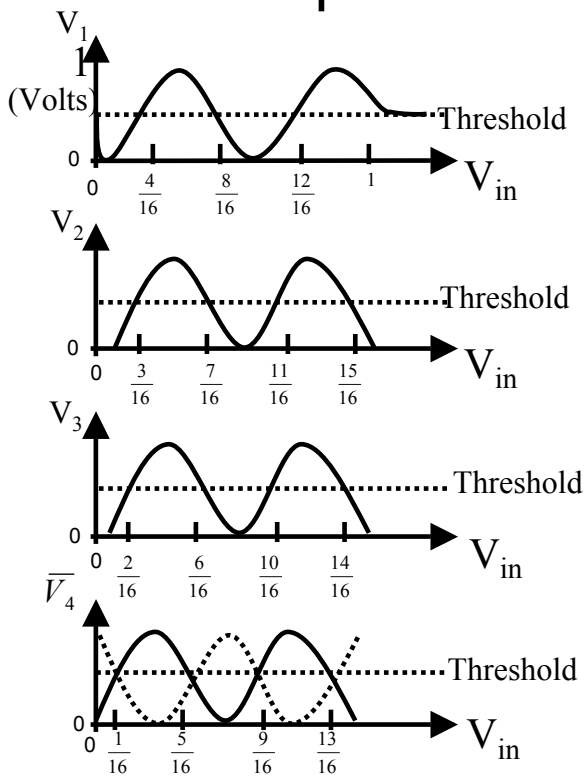
- Using V_1 folding block to also determine the top two MSB_s
  - 2-bit MSB ADC can be eliminated



Folding ADC(Cont.)

- Example3: A 4-bit ADC with a folding rate of four and an interpolate-by-two technique
  - folding + interpolating
  - smaller input loading compared to examples 1 and 2





Folding ADC(Cont.)

- Some Important Points of Folding ADC
  - Output signal from a folding block is at a much higher frequency than the input signal.

    Frequency of folding curve = Input frequency x Folding rate

    This multiplying effect limits the practical folding rate used in high frequency converters.
  - Differential circuits are almost always used in practical implementation.

Pipelined ADC

- The two-step ADC architecture can be generalized to multiple stages

  e.g. 1-bit/stage, 2-bit/stage——— without digital error correction

  1.5-bit/stage, 2-bit/stage, 2.5-bit/stage, ...etc.

  ——— with digital error correction

  In general, 1.5-bit/stage is the optimum with respect to speed, area and power.

- To increase conversion speed, pipelined structure should be incorporated. Once the first stage completes its work, it does not sit idle while the remaining lower bits are found, but immediately starts work on the next input sample.

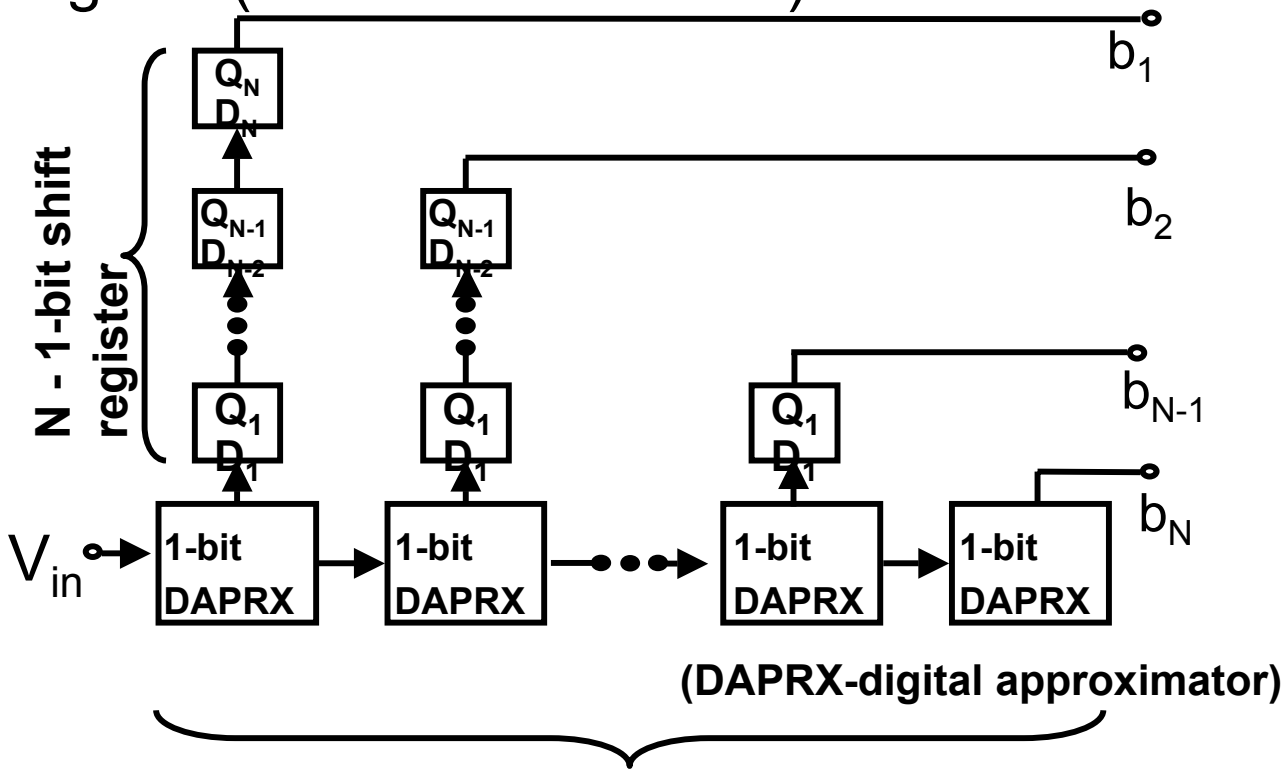
Pipelined ADC (cont.)

- It takes N clock cycles to process each input signal (i.e., latency is N).
- Conversion rate equals clock rate
- Circuit complexity is proportional to N.
- Small area
- Current state-of-art is 12 to 15 bits for pipelined ADC with digital error correction at 50 to 100 MHz

  The speed is expected to go up substantially in the near future.

Pipelined ADC (cont.)

- Block diagram (No error correction)



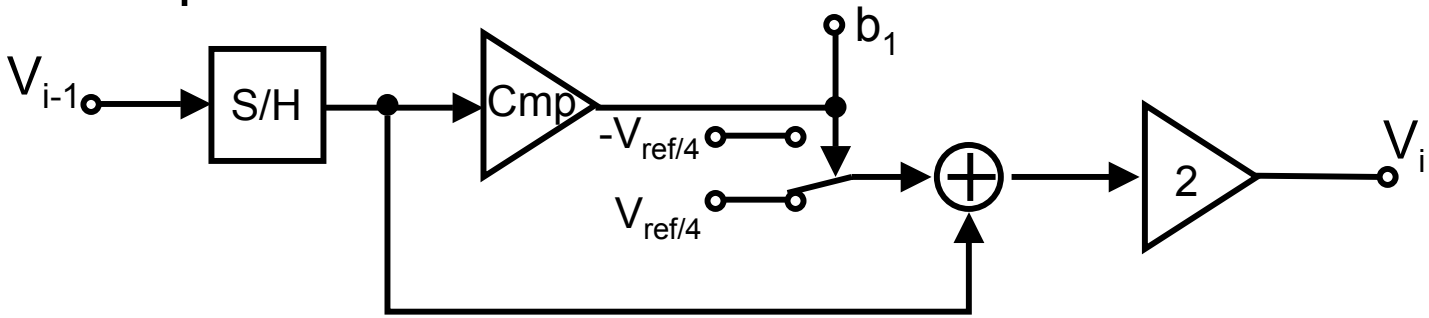
– For a signed conversion, the input voltage is compared to 0V. If $V_{in}>0$,then $V_{out}=2V_{in}-(V_{ref}/2)$, and Bout=1. Otherwise,$V_{out}=2V_{in}+(V_{ref}/2)$, and $B_{out}=0$

Pipelined ADC (cont.)

– digital approximator (DAPRX)

Each DAPRX contains an S/H to store the input signal. This S/H allows the proceeding DAPRX to be immediately used to process its next input signal before the succeeding DAPRX has finished.

1-bit example

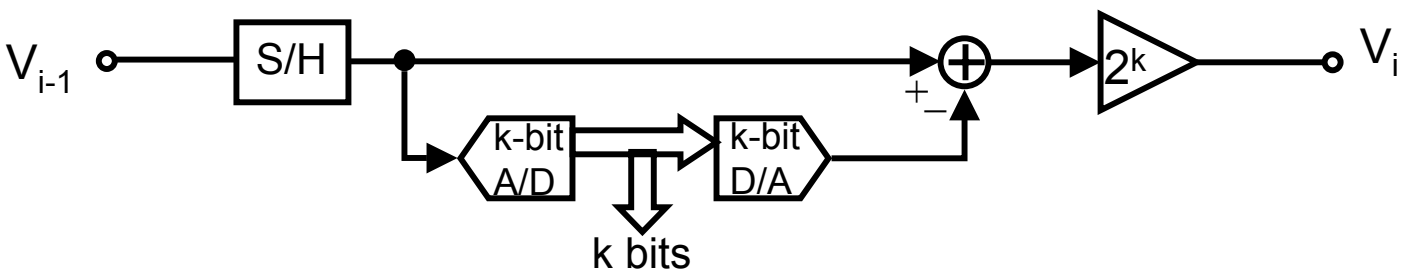


The ith S/H can be incorporated into the (i-1)th gain-of-two amplifiers except for the first stage of the pipelined ADC.

Pipelined ADC (cont.)

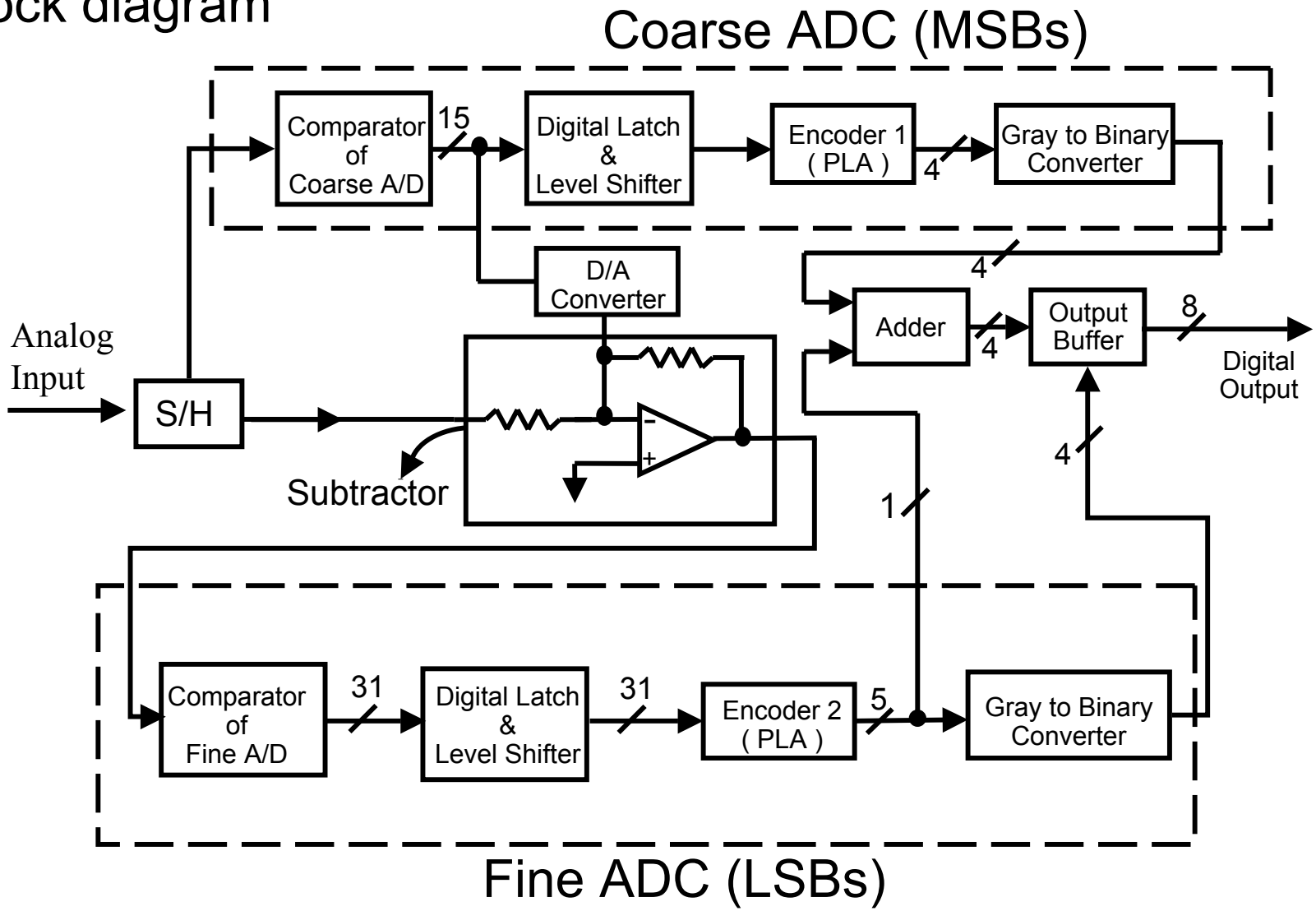
- Implementation of a multi-bit
  - Digital error correction can be added similar to that for a two-step ADC.
  - Major limitation on the accuracy is the gain amplifier, especially in the first few stages.

    ⇨ Gain is taken smaller for the first stages which makes high-speed amplifier design considerably easier.
  - SC implementation of S/H gain amplifier requires double-poly process(in general). Sometimes, double metal is ok if large C is not required.



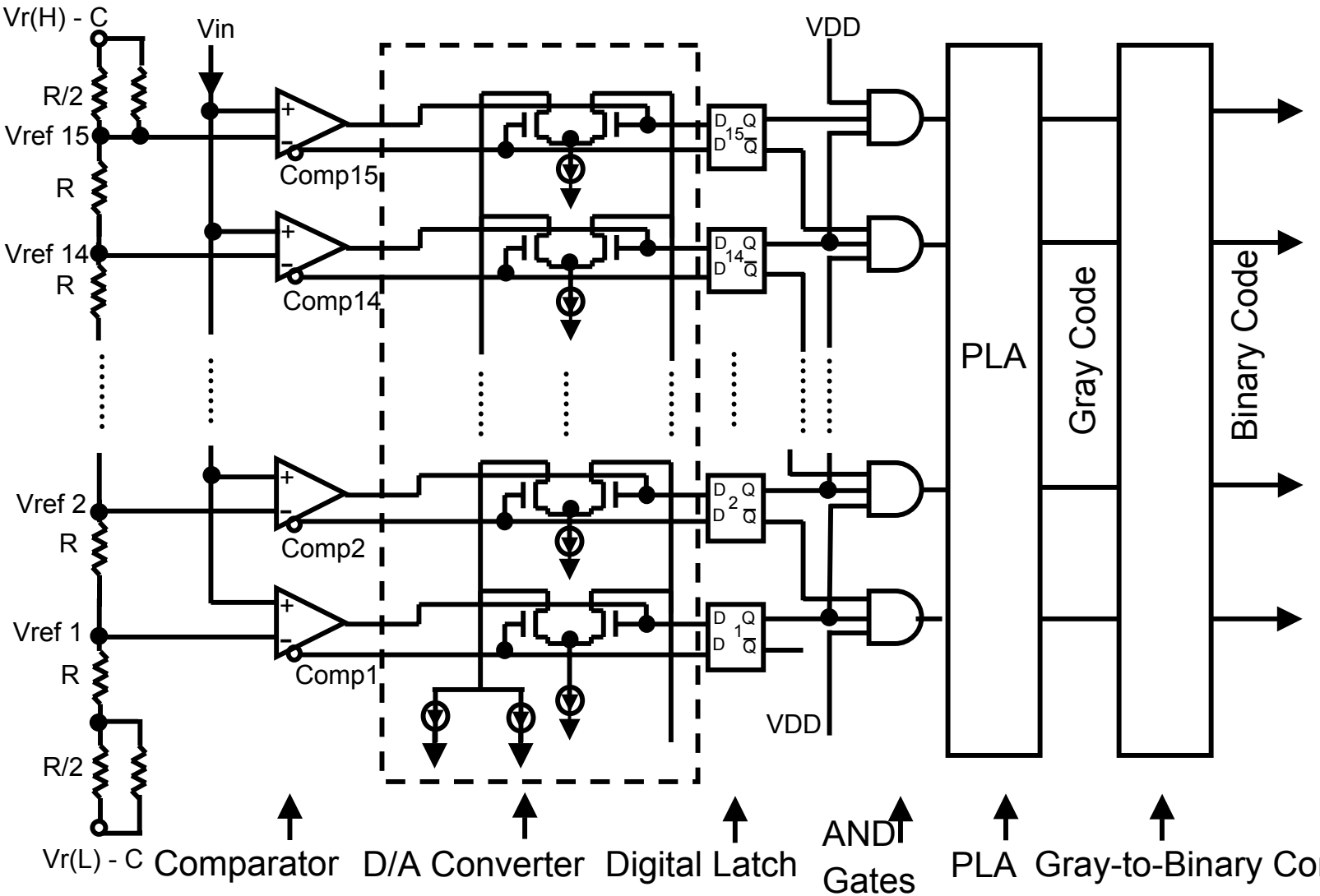
Case study 1: Implementation of an 8-bit Two-Step (or Subrange) ADC

- Block diagram



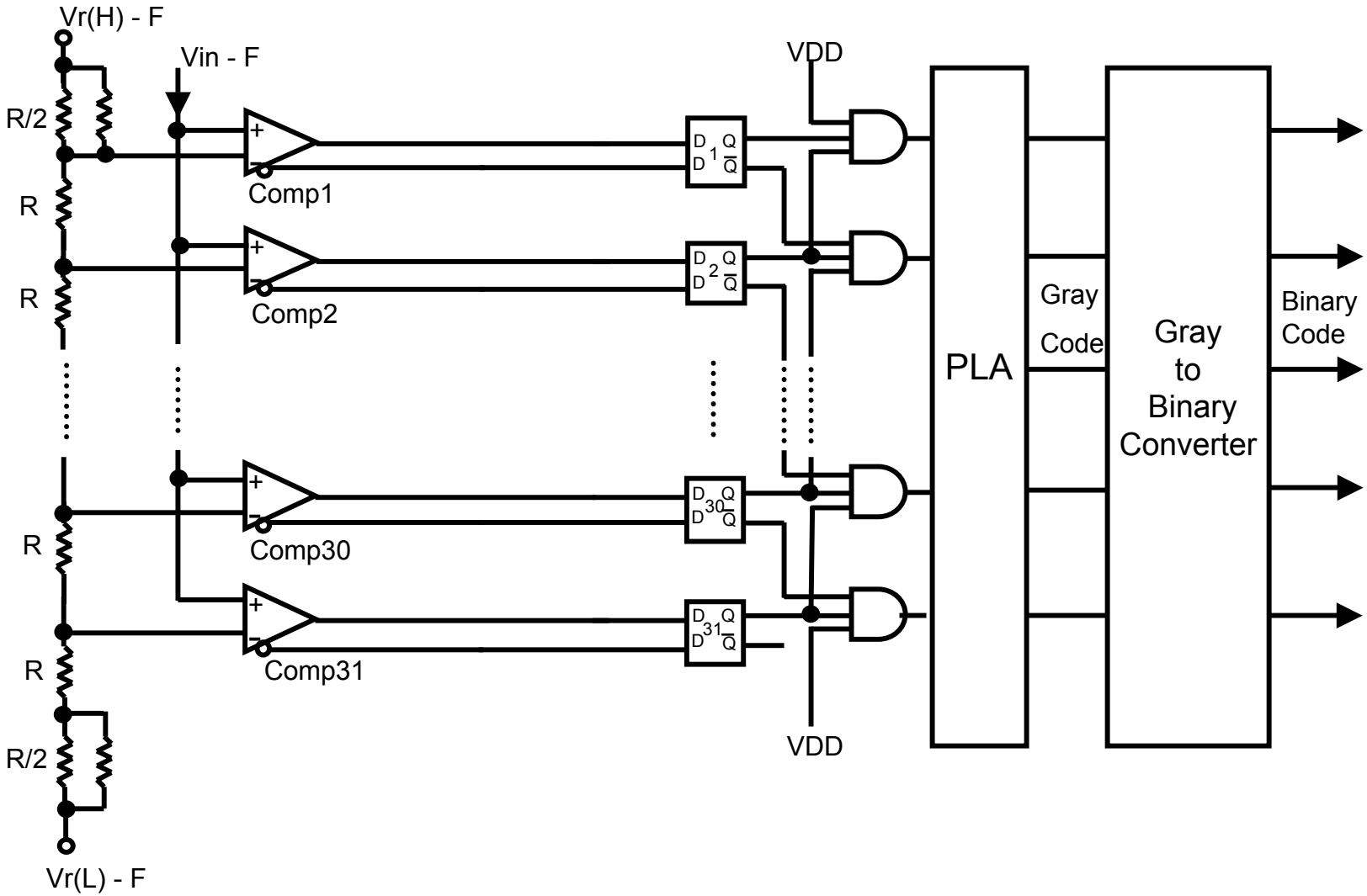
Case study 1: Implementation of an 8-bit Two-Step (or Subrange) ADC (cont.)

- Coarse ADC structure



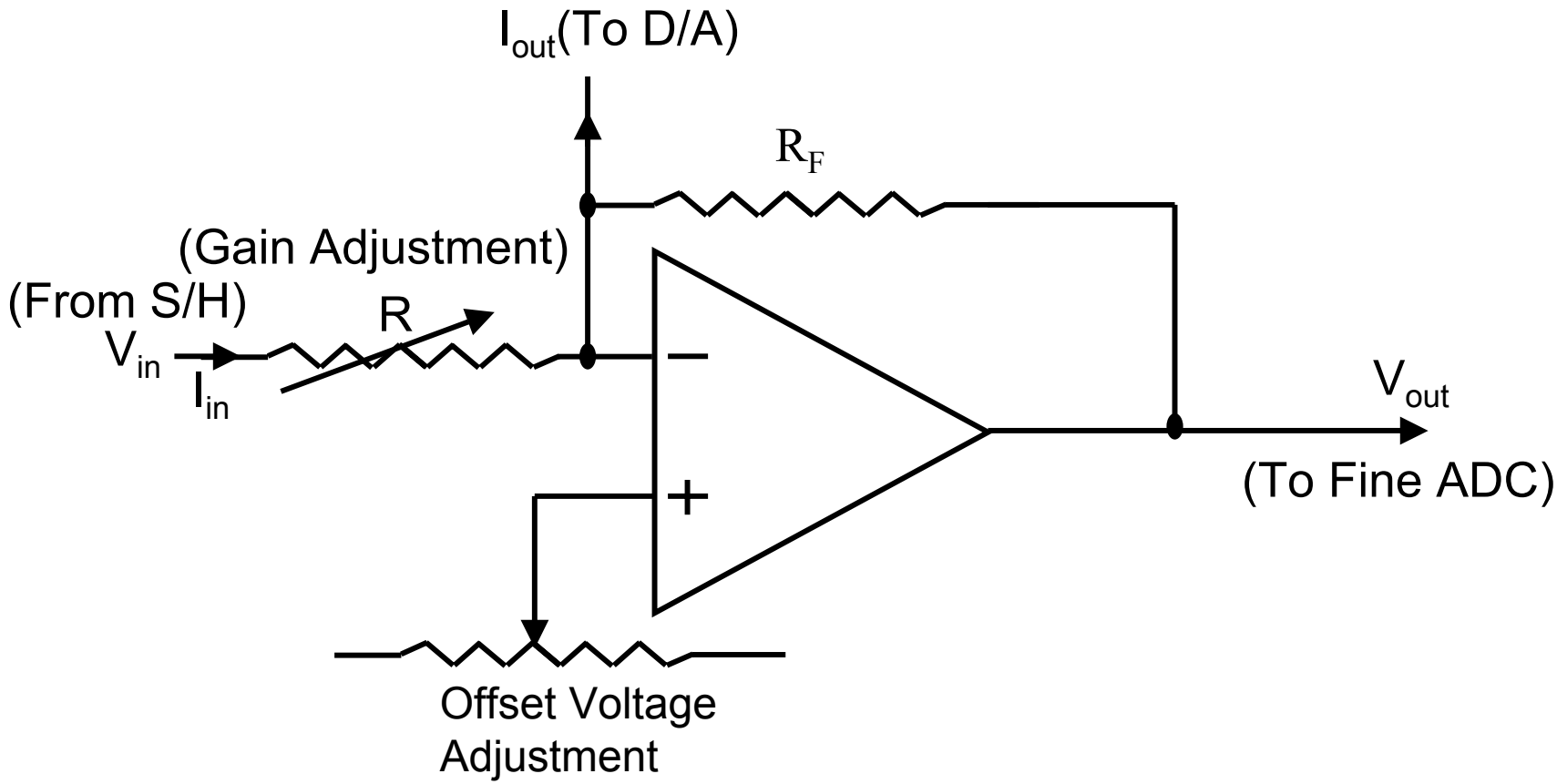
Case study 1: Implementation of an 8-bit Two-Step (or Subrange) ADC (cont.)

- Fine A/D structure



Case study 1: Implementation of an 8-bit Two-Step (or Subrange) ADC (cont.)

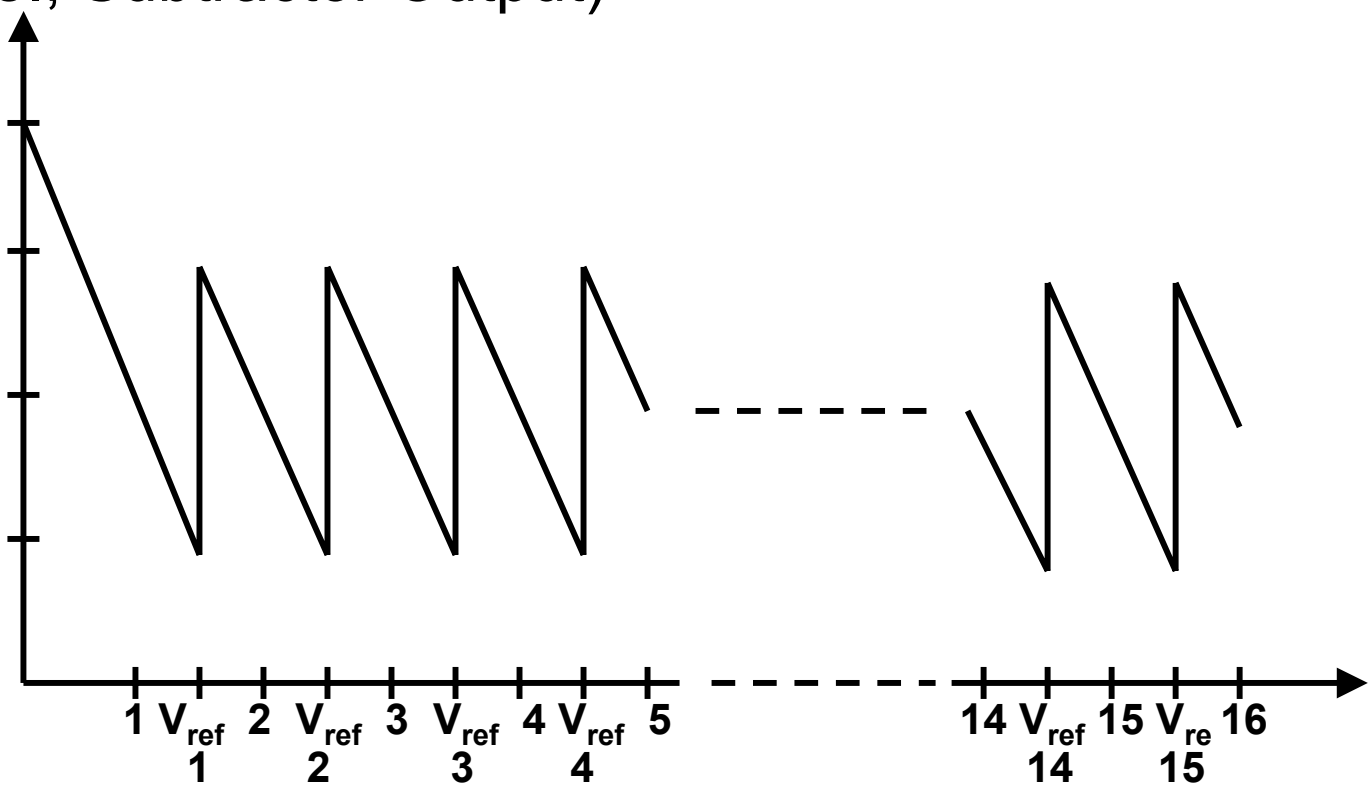
- Subtractor



Case study 1: Implementation of an 8-bit Two-Step (or Subrange) ADC (cont.)

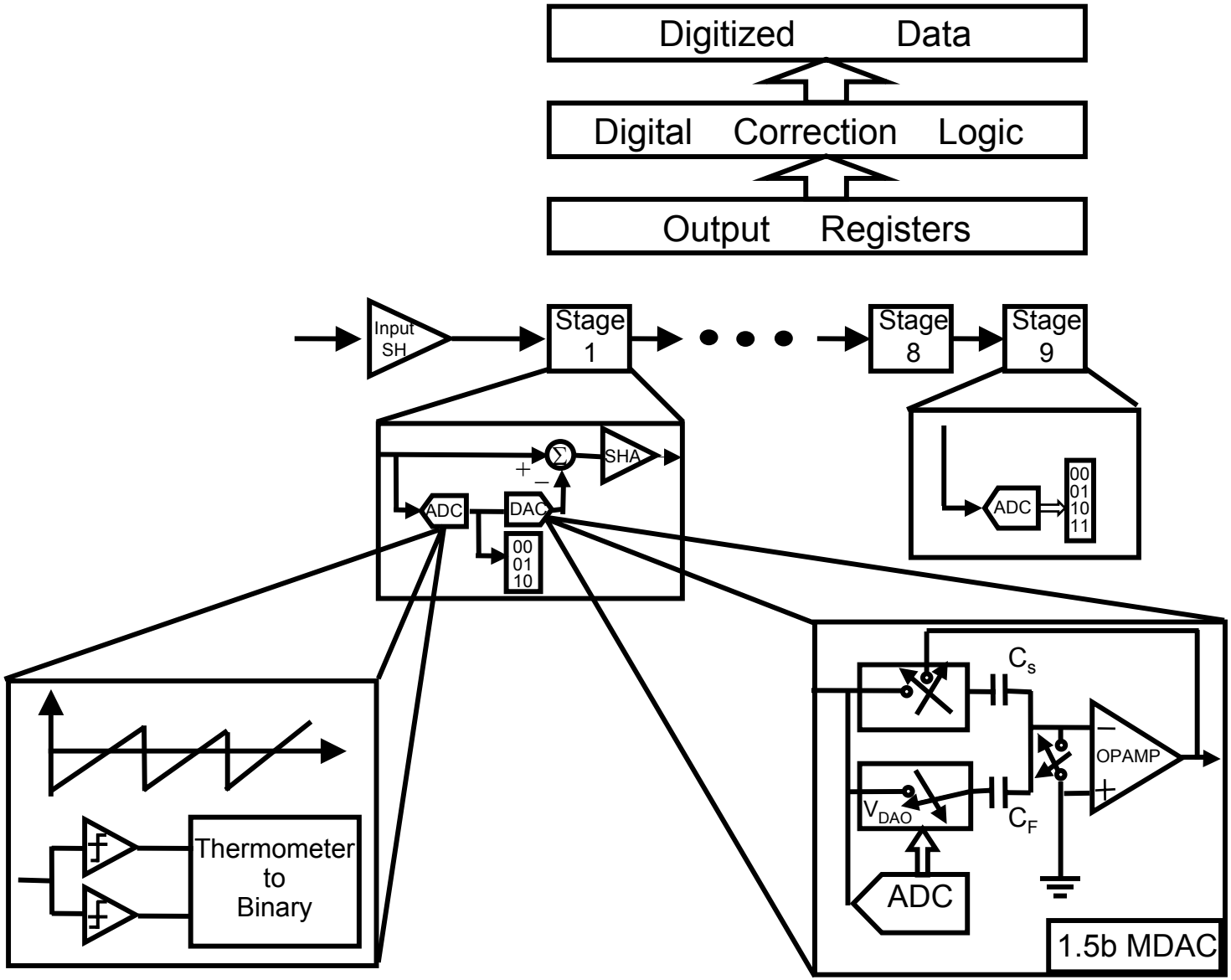
- Input relation between coarse and fine (i.e., residue plot)

Input Voltage To Fine ADC
(i.e., Subtractor Output)



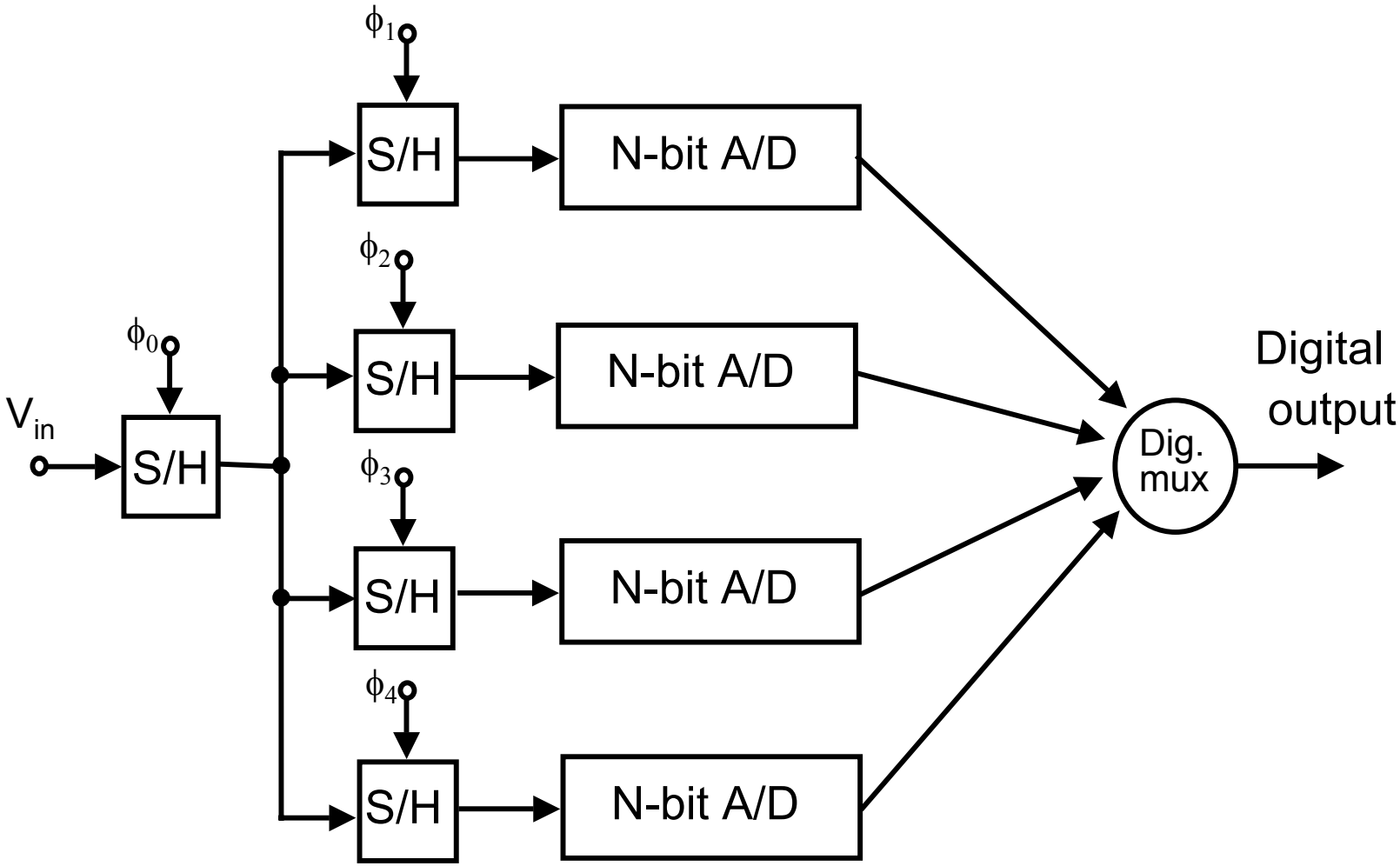
Input Voltage To Coarse ADC (i.e.S/H Output)

Case Study 2: 1.5b/Stage Pipeline Architecture

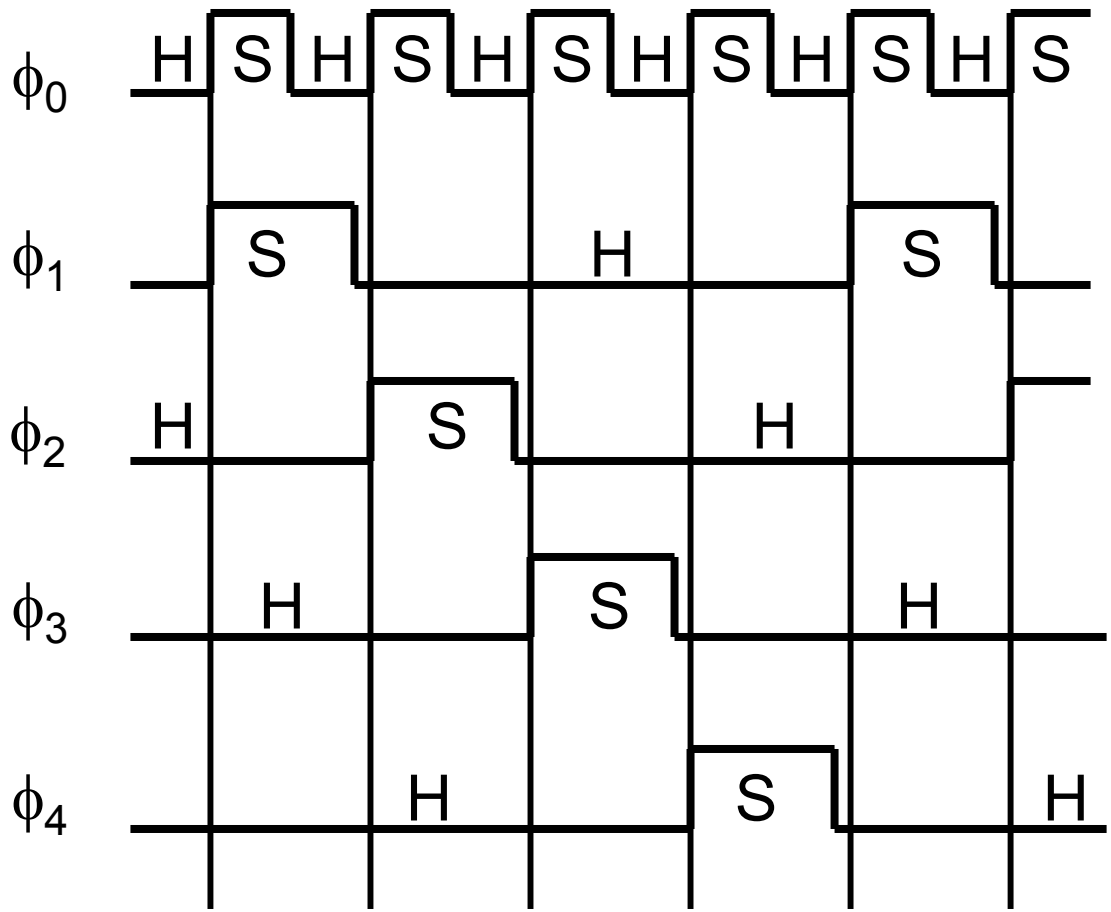


Time-Interleaved ADC

- Ultra-high speed is possible using this approach
- Operating many ADCs in parallel



Time-Interleaved ADC (cont.)



Time-Interleaved ADC (cont.)

- The four ADCs operate at one-quarter the rate of the input sampling frequency .
- The input S/H making use of ϕ_0 is critical, while the remaining four S/H converters can have considerable jitter since the signal is already sampled at that point. Sometimes, the input S/H is realized in different technology, such as GaAs, while the S/H circuits could be realized in silicon.
- It is also essential that the channels are extremely well matched, as mismatches will produce tones at fs/m when there are m channels.

  Such nonideal behavior can be disastrous for many applications since the tone may reside well within the frequency of interest.