A/D Converters

Nyquist - Rate A/D Converters

- ADCs can be roughly divided into three categories

Low-to-Medium Speed, High Accuracy		Medium Speed, Medium Accuracy		High Speed, Low-to-Medium Accuracy	
Integrating	(1)	Successive approximation	(1,2,3,4)	Flash	(1,2,3,4)
Delta-Sigma	(2,3,4)	Algorithmic/ cyclic	(1)	Two-step	(1,2,3,4)
		Delta-Sigma	(2,3,4)	Interpolating	(2,3)
				Folding	(2)
				Pipelined	(2,3,4)
				Time-interleaved	(3,4)

- Popular around: (1) Before 1990
  (2) 1990~2005
  (3) 2005~2015
  (4) After 2015

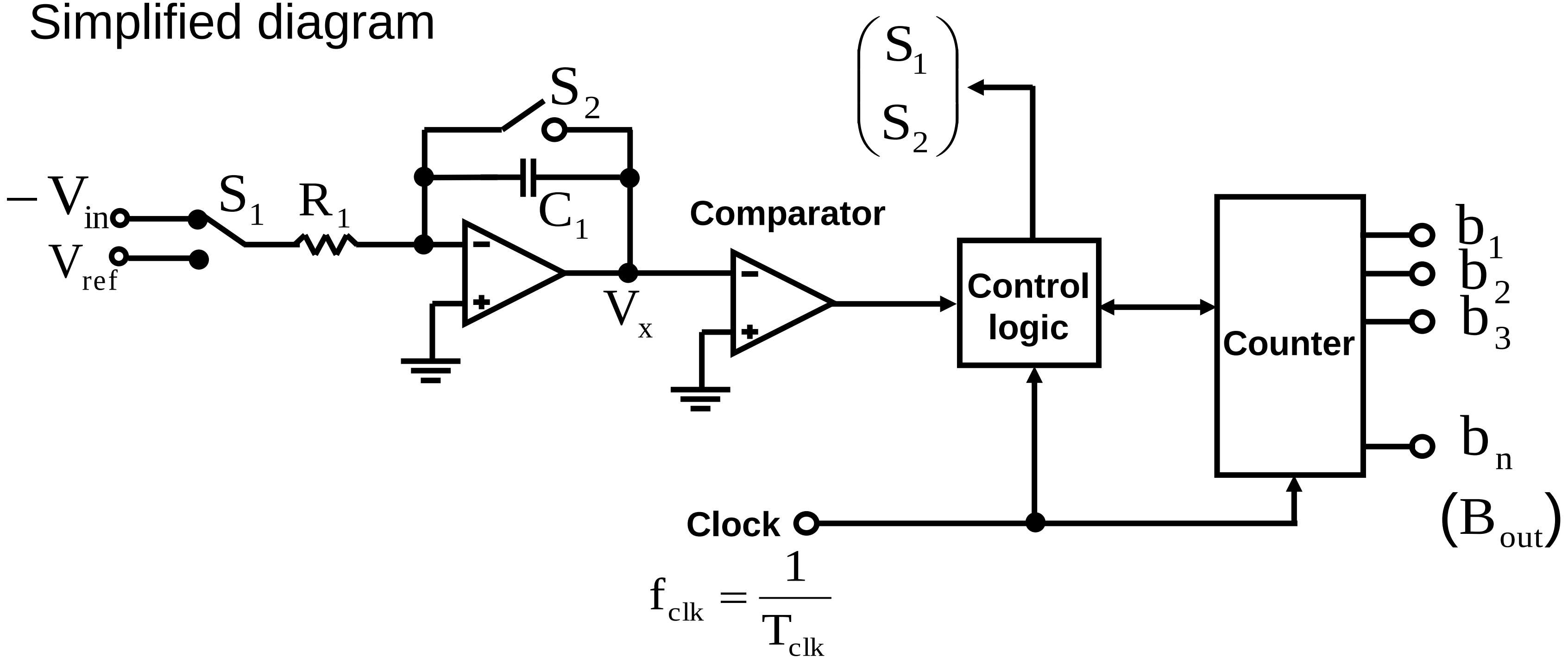
Integrating ADC (or Dual-Slope ADC)

- 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

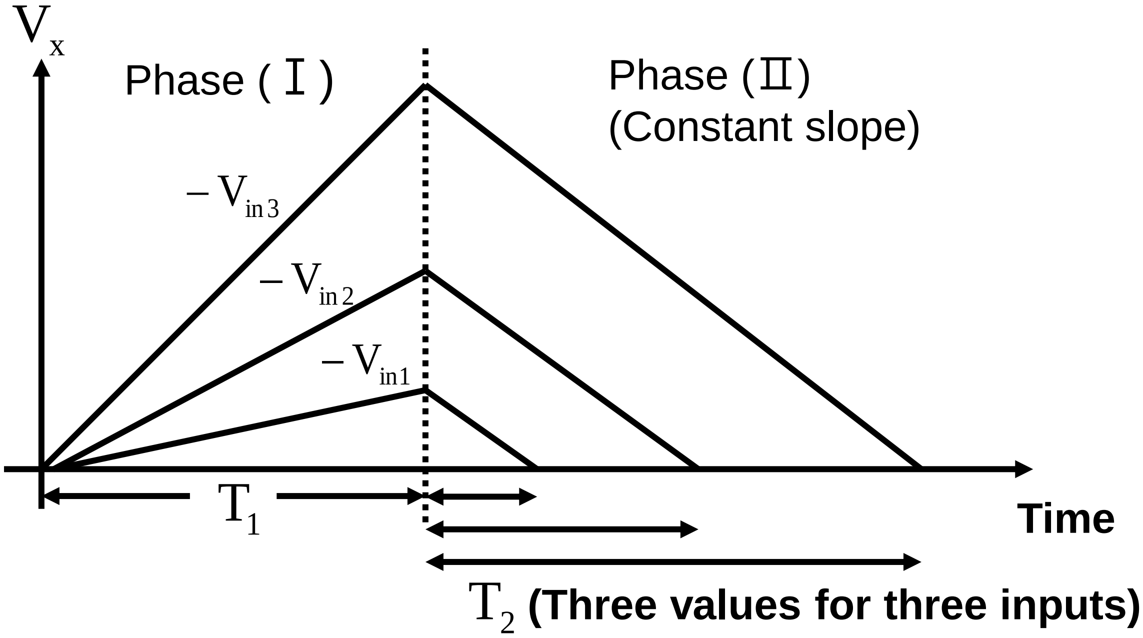


Integrating ADC (or Dual-Slope ADC) (Cont.)

- 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, $V_x(T_1) = \int_0^{T_1} \frac{V_{in}}{R_1}\frac{1}{C_1}dt = \frac{V_{in}T_1}{R_1C_1}$

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



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

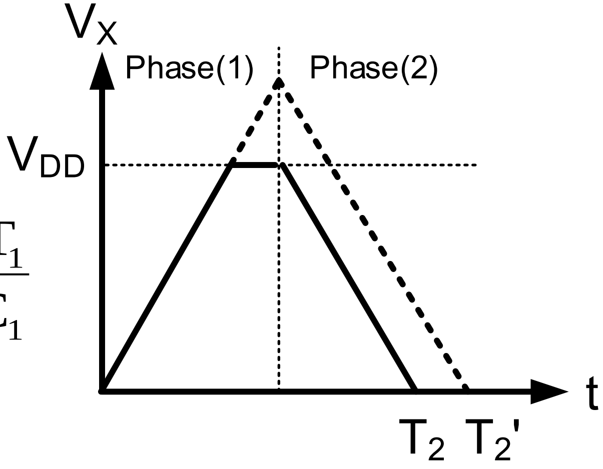
Integrating ADC (or Dual-Slope ADC) (Cont.)

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

  $B_{out} = b_1 2^{-1} + b_2 2^{-2} + \dots + b_N 2^{-N}$

  and we have

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



$$V_x(t) = -\int_{T_1}^{t} \frac{V_{ref}}{R_1C_1} d\tau + V_x(T_1) = -\frac{V_{ref}}{R_1C_1}(t - T_1) + \frac{V_{in}T_1}{R_1C_1}$$

  Since $V_x(t) = 0$, when $t = T_1 + T_2$

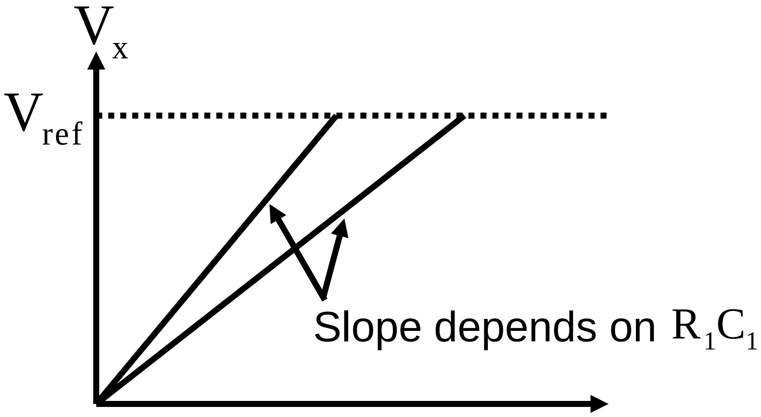
$$\frac{-V_{ref}T_2}{R_1C_1} + \frac{V_{in}T_1}{R_1C_1} = 0 \quad \Rightarrow T_2 = T_1\left(\frac{V_{in}}{V_{ref}}\right)$$

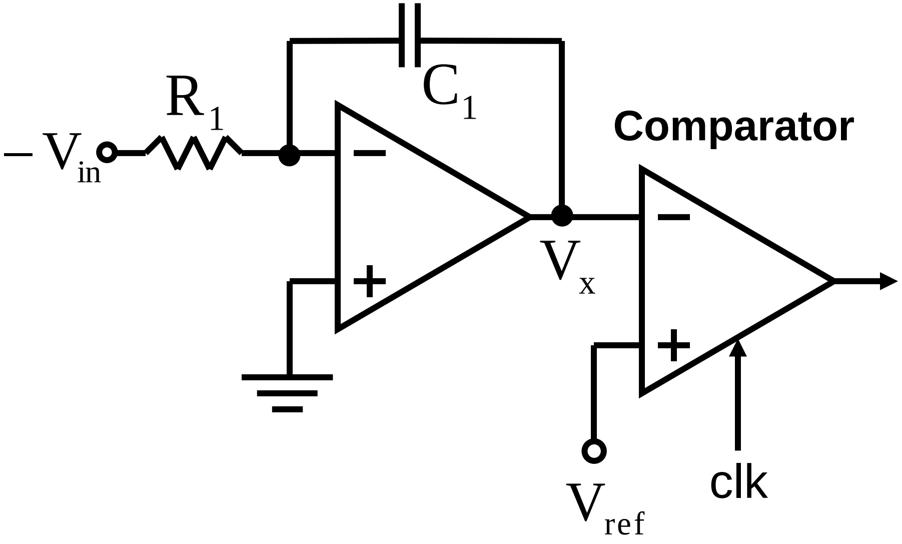
$$\Rightarrow B_{out} = b_1 2^{-1} + b_2 2^{-2} + \cdots + b_N = \frac{V_{in}}{V_{ref}} = \frac{T_2}{T_1}$$

- 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.

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

- 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)

  - ◆ Measure zero input first , then memorize its digital output, B_x
  - ◆ Measure full-scale DC signal, then memorize its digital output, B_y

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

- Quite slow

  $2 \cdot 2^N$ 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.

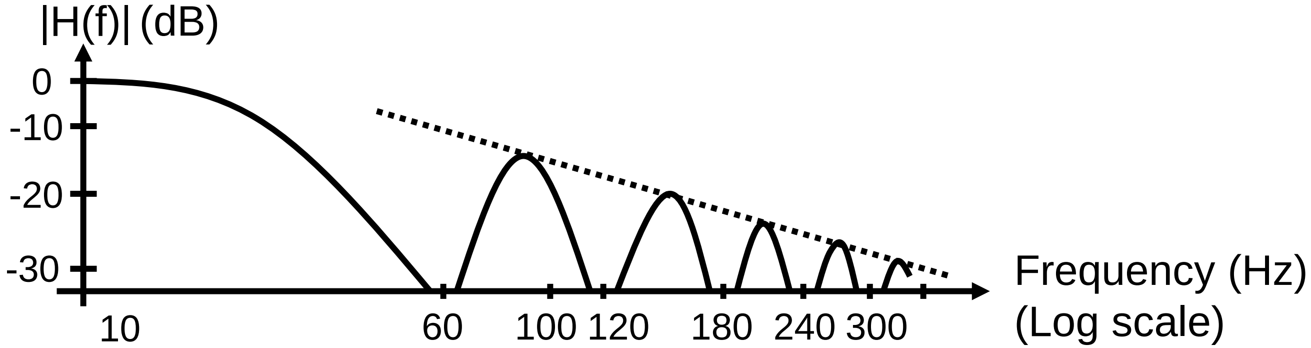
Integrating ADC (or Dual-Slope ADC) (Cont.)

- 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 $V_{in}(t) = V_{in}cos(2\pi ft + \theta)$, where V_{in} are arbitrary magnitude

$$V_x(T_1) = \int_0^{T_1} \frac{V_{in}\cos(2\pi ft)}{R_1C_1}dt = \frac{V_{in}\sin(2\pi ft)}{2\pi fR_1C_1}\bigg|_0^{T_1} = \frac{V_{in}T_1}{R_1C_1}\frac{\sin(2\pi fT_1)}{2\pi fT_1}$$

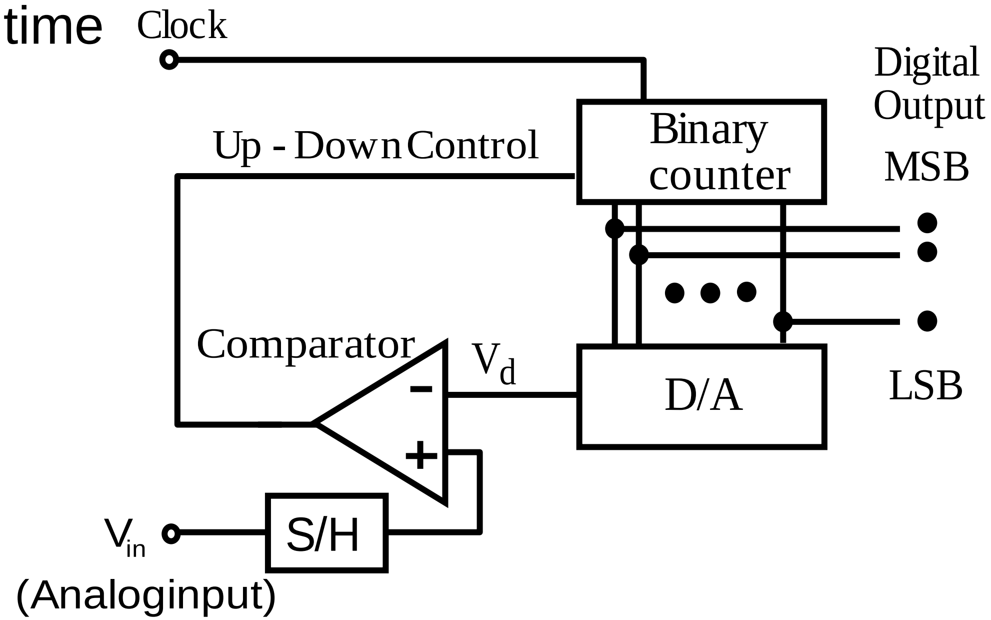
$$V_x(T_1) \approx H(f) \times \frac{V_{in}T_1}{R_1C_1} \quad \text{when } fT_1 \text{ approaches } 1, 2, 3, \ldots$$

  - Filter transfer function $|H(f)| = \left|\frac{\sin(\pi fT_1)}{\pi fT_1}\right| = |sinc(\pi fT_1)|$
- An example to filter out power line noise, especially 60Hz
  - T_1 is chosen to be an integer multiples of 16.67ms.
  - 60Hz, 120Hz, 180Hz, …… are suppressed.



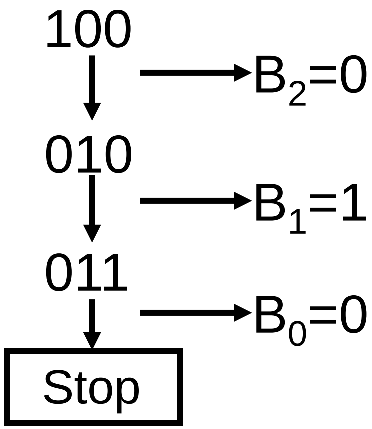
Successive Approximation (SA) ADC

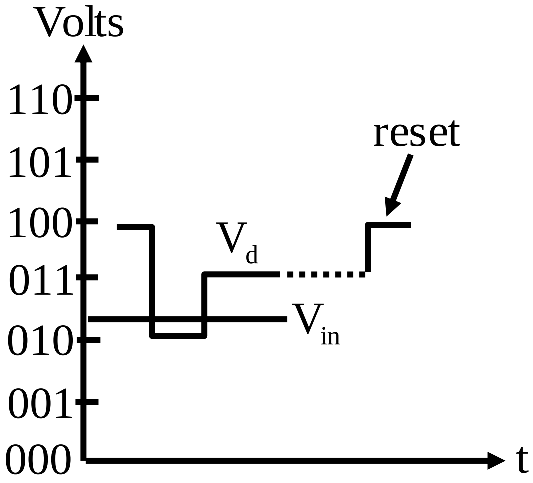
- Reasonably quick conversion time
  Moderate circuit complexity
- Binary search to determine the closest digital word to match analog input, N clock cycles to complete an N-bit conversion



- A 3-bit unipolar example

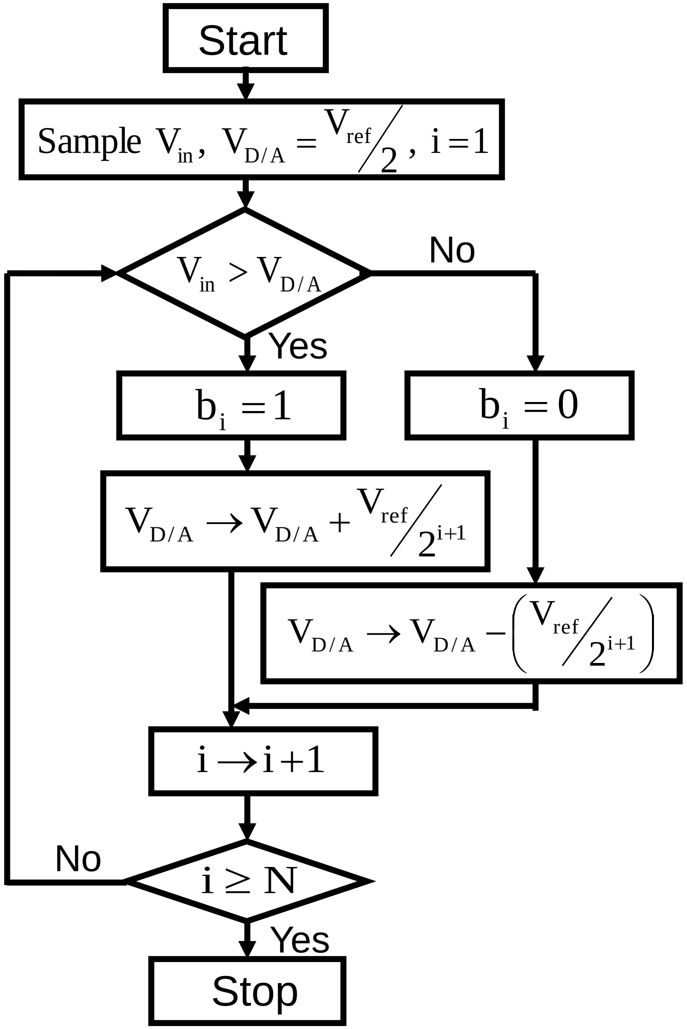
compare





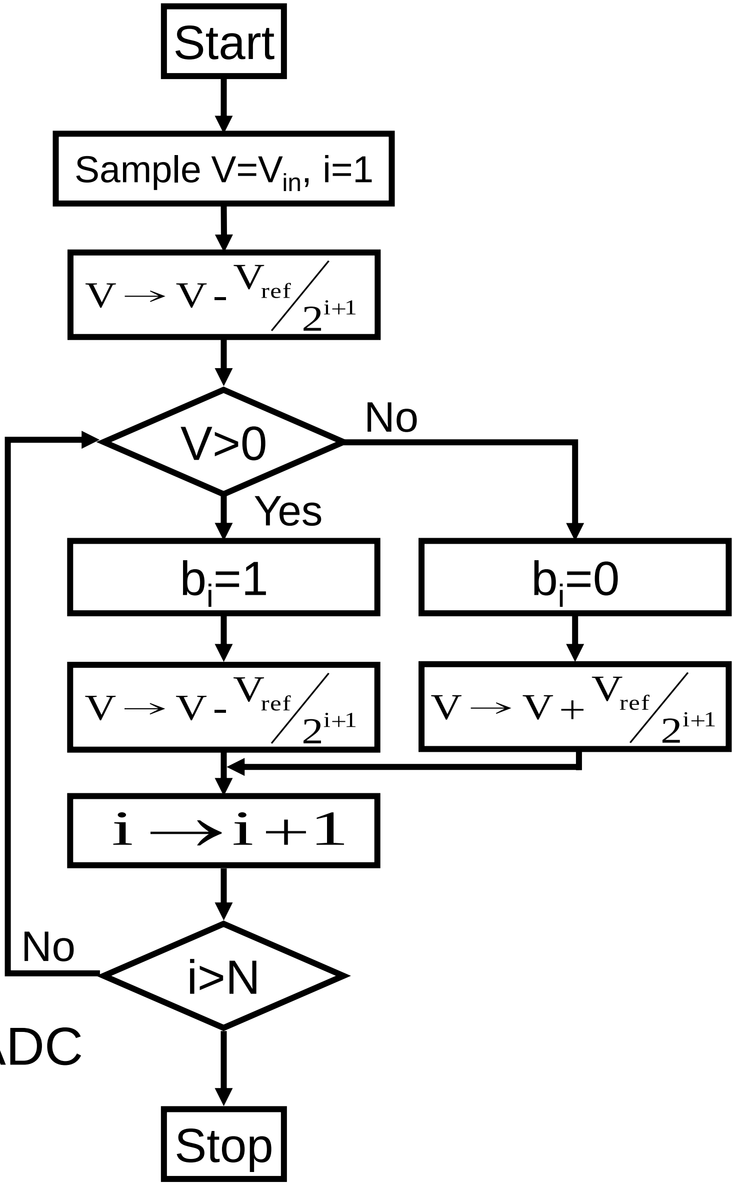
Successive Approximation (SA) ADC (Cont.)

- Flow graph for SA ADC
  - Unipolar example : Input range : 0 ~ V_{ref}


Start
$\text{Sample } V_{in},\ V_{D/A} = V_{ref}/2,\ i = 1$
$V_{in} > V_{D/A}$
No
Yes
$b_i = 1$
$b_i = 0$
$V_{D/A} \rightarrow V_{D/A} + V_{ref}/2^{i+1}$
$V_{D/A} \rightarrow V_{D/A} - \left(V_{ref}/2^{i+1}\right)$
$i \rightarrow i+1$
No
$i \geq N$
Yes
Stop

Modified Successive Approximation (SA) ADC

- Flow graph modified from that in p.12-11
  - No need for a separate DAC
    S/H, DAC, and difference portion of the comparator are all combined into a single circuit.
    - The error V equals the difference between input V_{in} and DAC output.
    - V is always compared to ground.
    - Charge-redistribution MOSFET ADC is one of the first switched-capacitor ADC using this approach.
  - Also called charge redistribution SA ADC

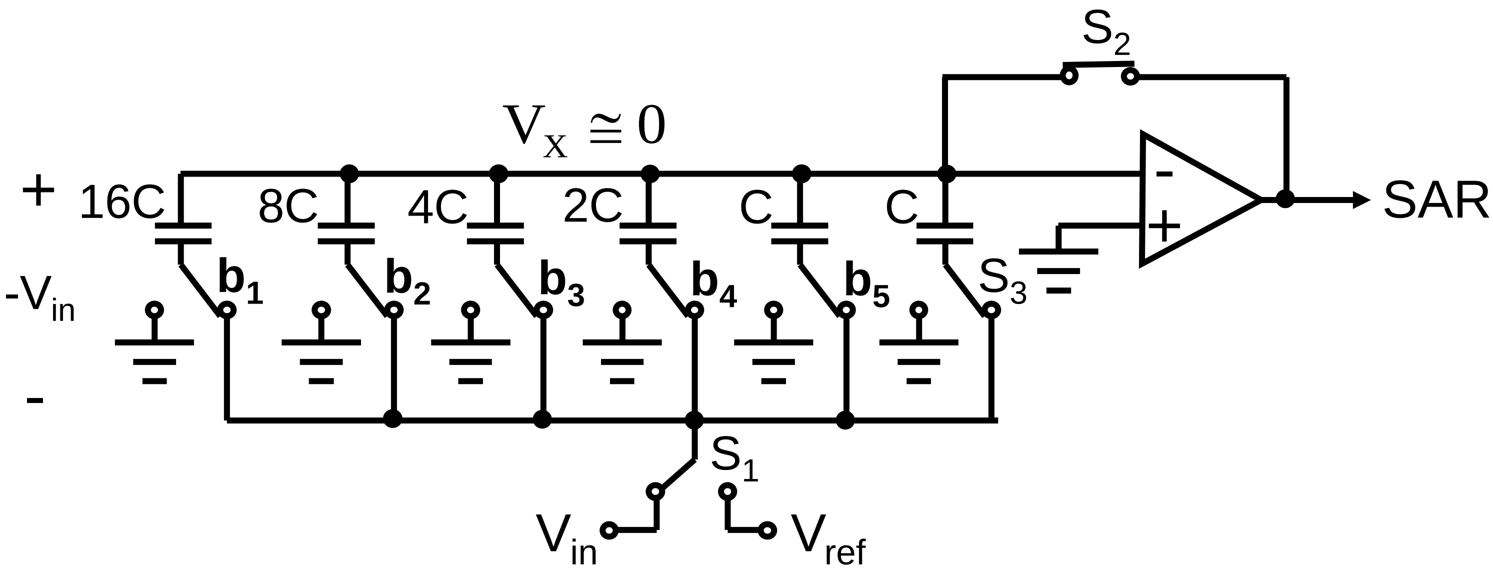


Start
Sample $V=V_{in}$, i=1
$V \rightarrow V - V_{ref}/2^{i+1}$
V>0
No
Yes
$b_i=1$
$b_i=0$
$V \rightarrow V - V_{ref}/2^{i+1}$
$V \rightarrow V + V_{ref}/2^{i+1}$
$i \rightarrow i+1$
No
i>N
Stop

Charge Redistribution SA ADC

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

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



Charge Redistribution SA ADC (Cont.)

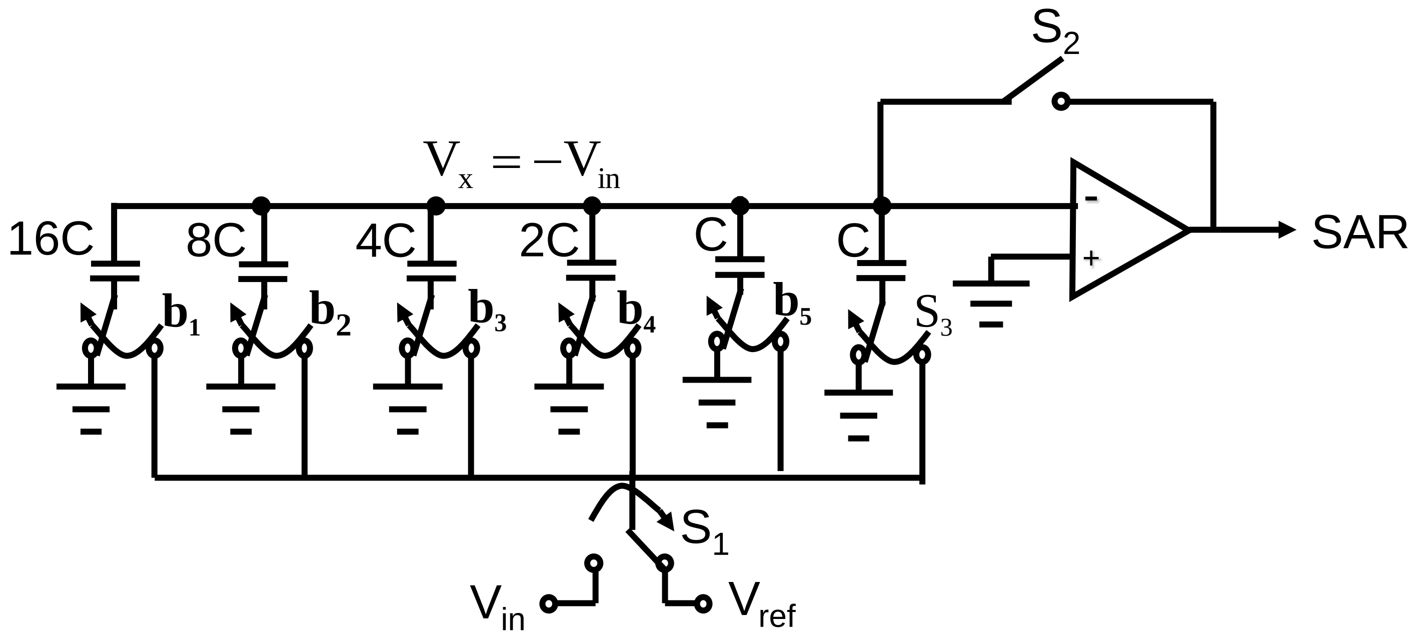
- Hold mode

  Comparator is taken out of reset.

  All capacitors are switched to ground.

  V_x : $-V_{in}$

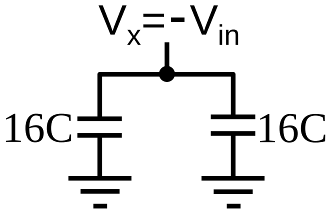
  V_{in} is held on the capacitor array



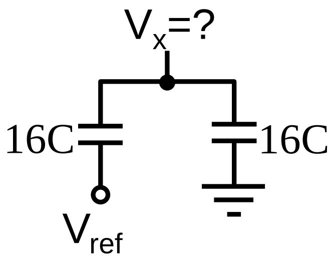
Charge Redistribution SA ADC (Cont.)

- Bit cycling

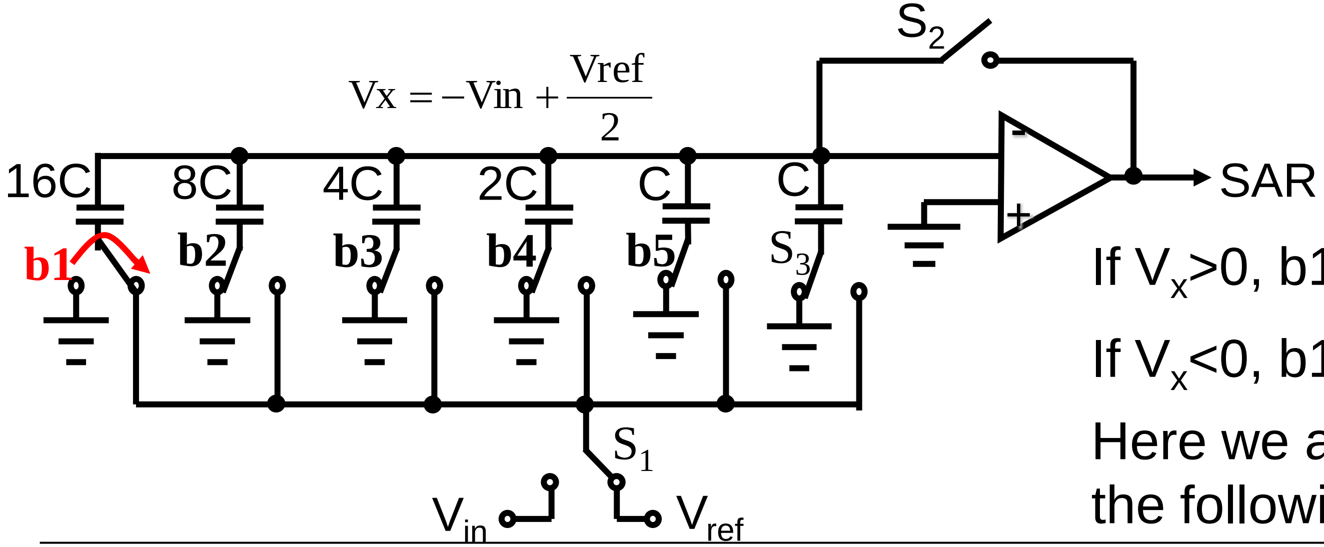
The largest capacitor is switched to V_{ref}



$$Q = 16C\cdot(-V_{in}) + 16C\cdot(-V_{in})$$
$$= -32C\cdot V_{in}$$



$$Q = 16C\cdot(V_x - V_{ref}) + 16C\cdot V_x$$
$$= 16C\cdot(2V_x - V_{ref}) = -32CV_{in}$$
$$\boxed{\Rightarrow V_x = -V_{in} + \frac{V_{ref}}{2}}$$



If V_x>0, b1 is determined to be 0

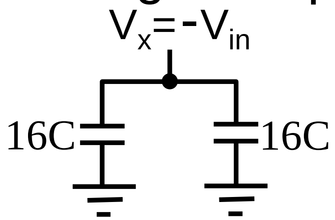
If V_x<0, b1 is determined to be 1

Here we assume b1=1 to explain the following operation

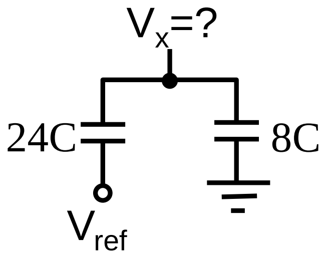
Charge Redistribution SA ADC (Cont.)

- Bit cycling (Assume b1 is already determined to be 1)

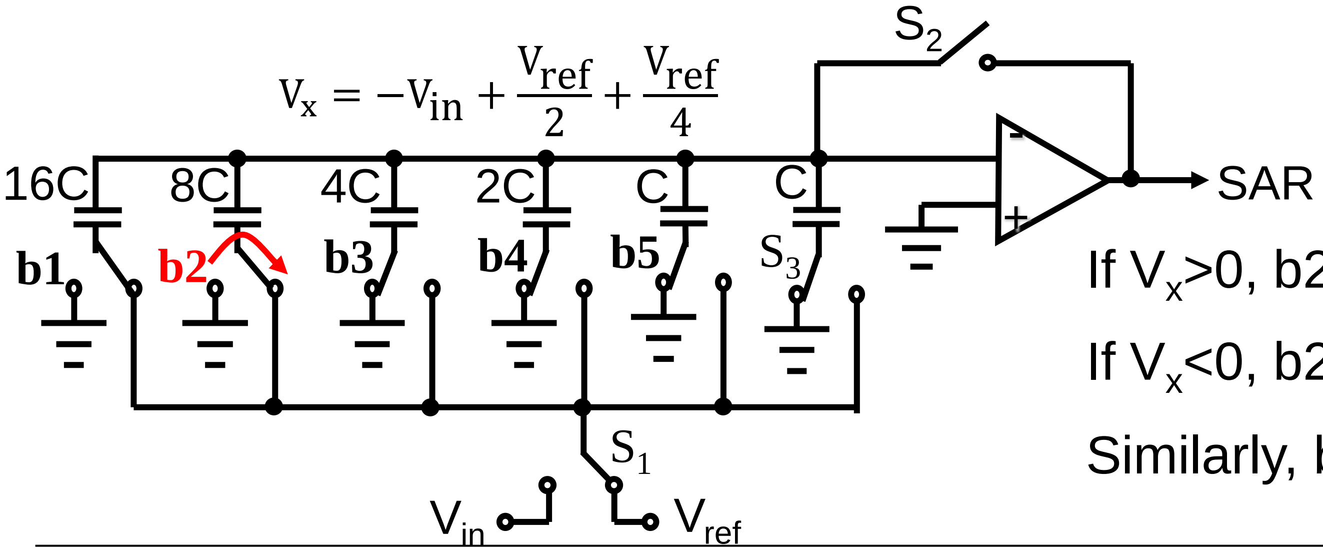
The second largest capacitor is switched to V_{ref}



$$Q = 16C \cdot (-V_{in}) + 16C \cdot (-V_{in})$$
$$= -32C \cdot V_{in}$$



$$Q = 24C \cdot (V_x - V_{ref}) + 8C \cdot V_x$$
$$= 32CV_x - 24CV_{ref} = -32CV_{in}$$
$$\boxed{\Rightarrow V_x = -V_{in} + \frac{3V_{ref}}{4} = -V_{in} + \frac{V_{ref}}{2} + \frac{V_{ref}}{4}}$$


$$V_x = -V_{in} + \frac{V_{ref}}{2} + \frac{V_{ref}}{4}$$

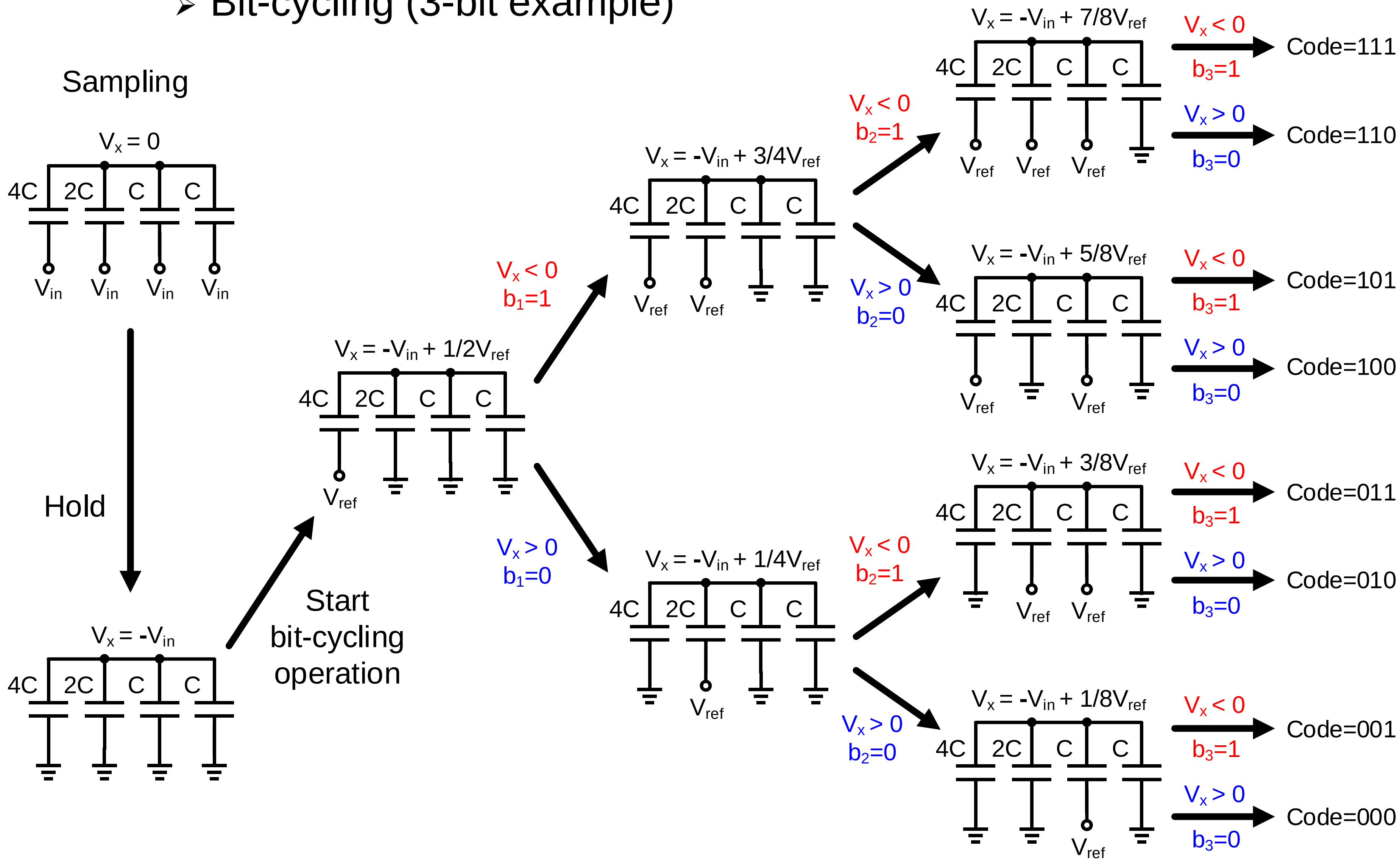
If V_x>0, b2 is determined to be 0

If V_x<0, b2 is determined to be 1

Similarly, b3~b5 can be obtained

Charge Redistribution SA ADC (Cont.)

- Bit-cycling (3-bit example)

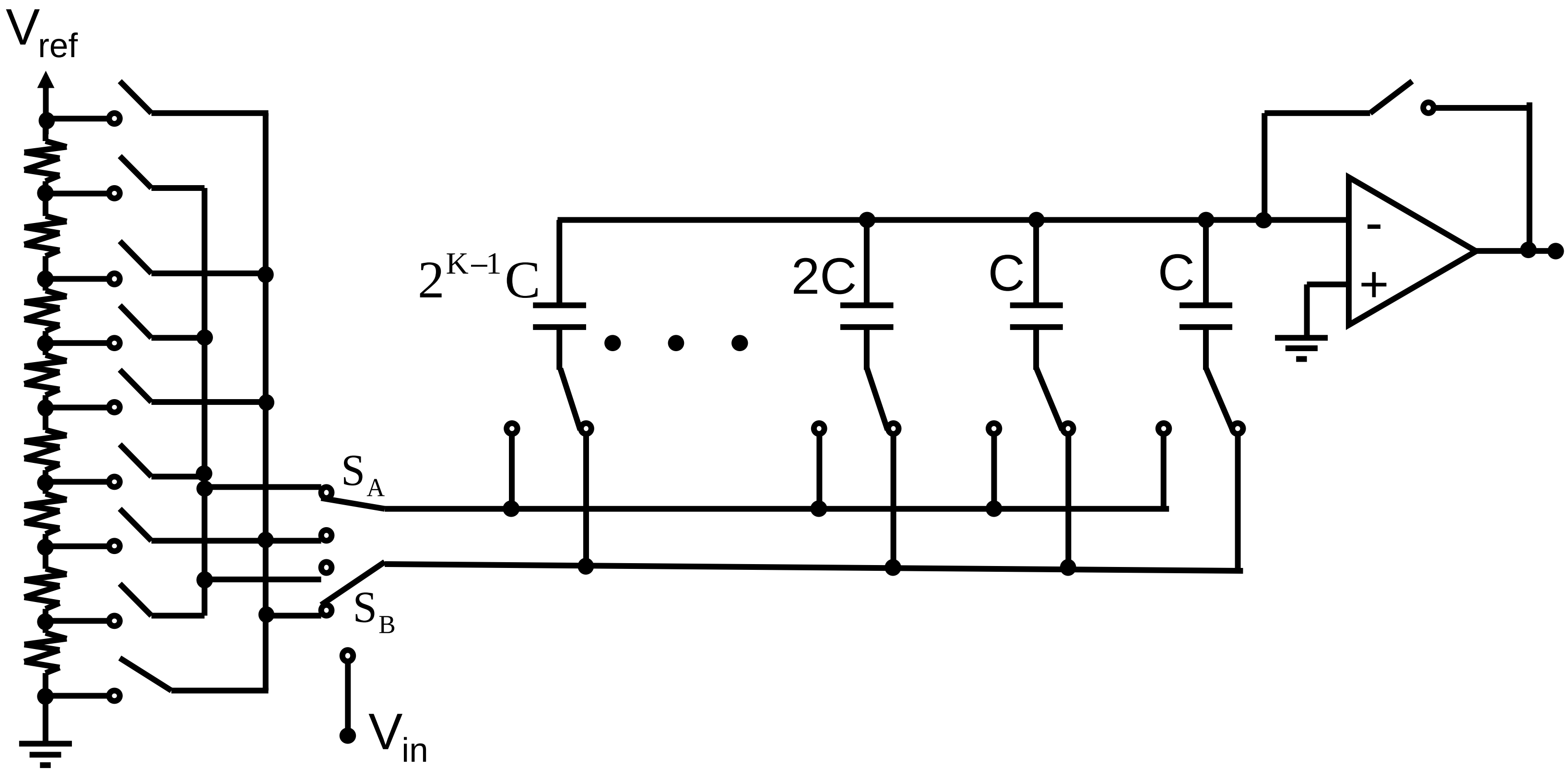


Charge Redistribution SA ADC (Cont.)

- 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 plates should be connected to V_{ref} side, not to comparator side, to minimize parasitic capacitance at node V_x. Although parasitic capacitance at V_x does not cause any conversion errors with an ideal comparator, it does attenuate V_x.

Resistor-Capacitor Hybrid ADC

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



Resistor-Capacitor Hybrid ADC (Cont.)

- 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.
- All of the capacitors are connected to the bus having the lower voltage.
- SA using the capacitor-array network
  - Starting with the largest capacitor, a capacitor is switched to the adjacent resistor-string node having a larger voltage.
  - If the comparator output is a 1, it is switched back and is a 0. Otherwise, the switch is left as is and is a 1.

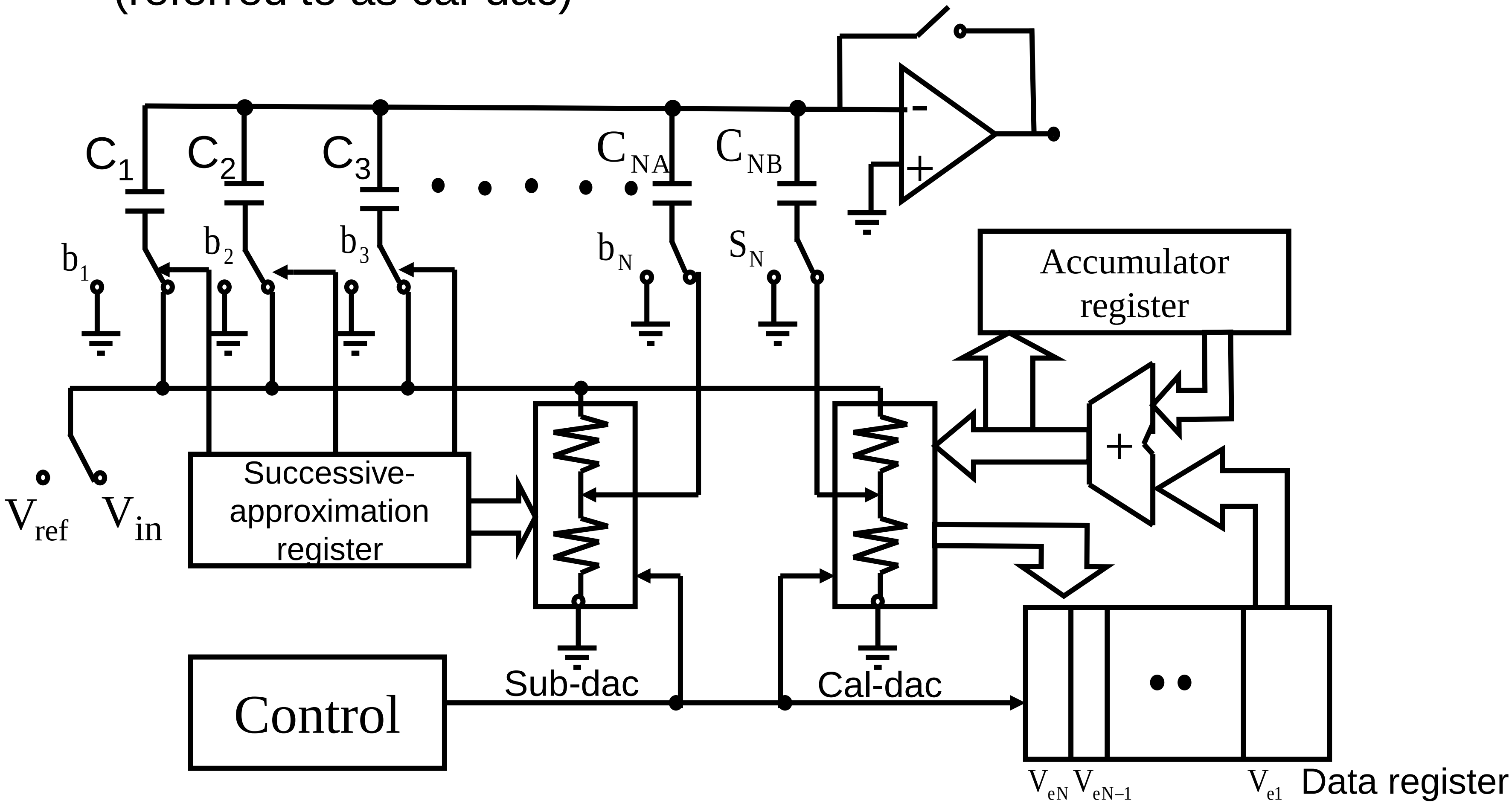
Charge-Redistribution with Error Correction

- 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
    - An additional capacitor and
    - 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)

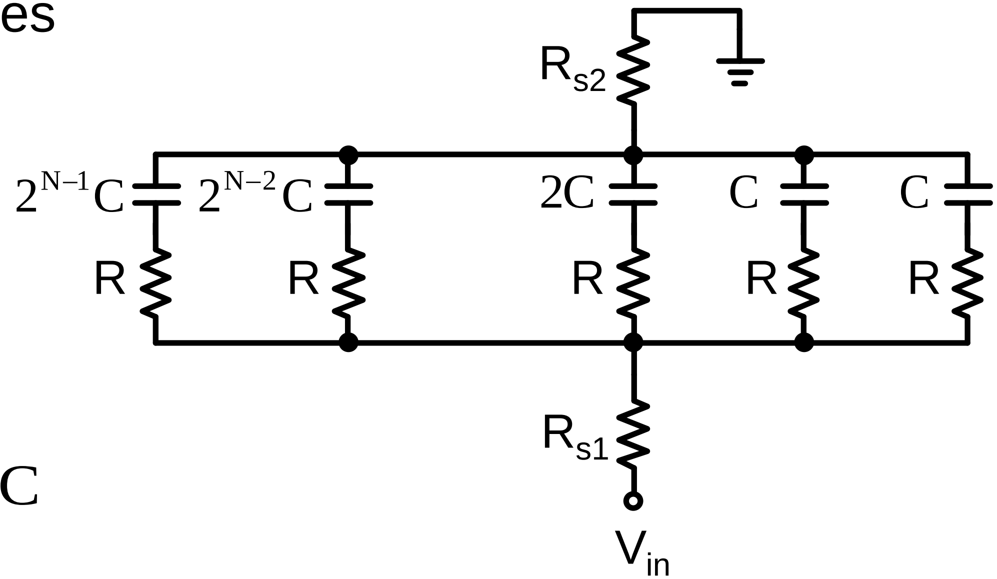


Charge-Redistribution with Error Correction (Cont.)

- Calibration
  - Measuring the errors of each capacitor, starting with the largest capacitor, 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}.

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

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

  - For better than 0.5LSB accuracy

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

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

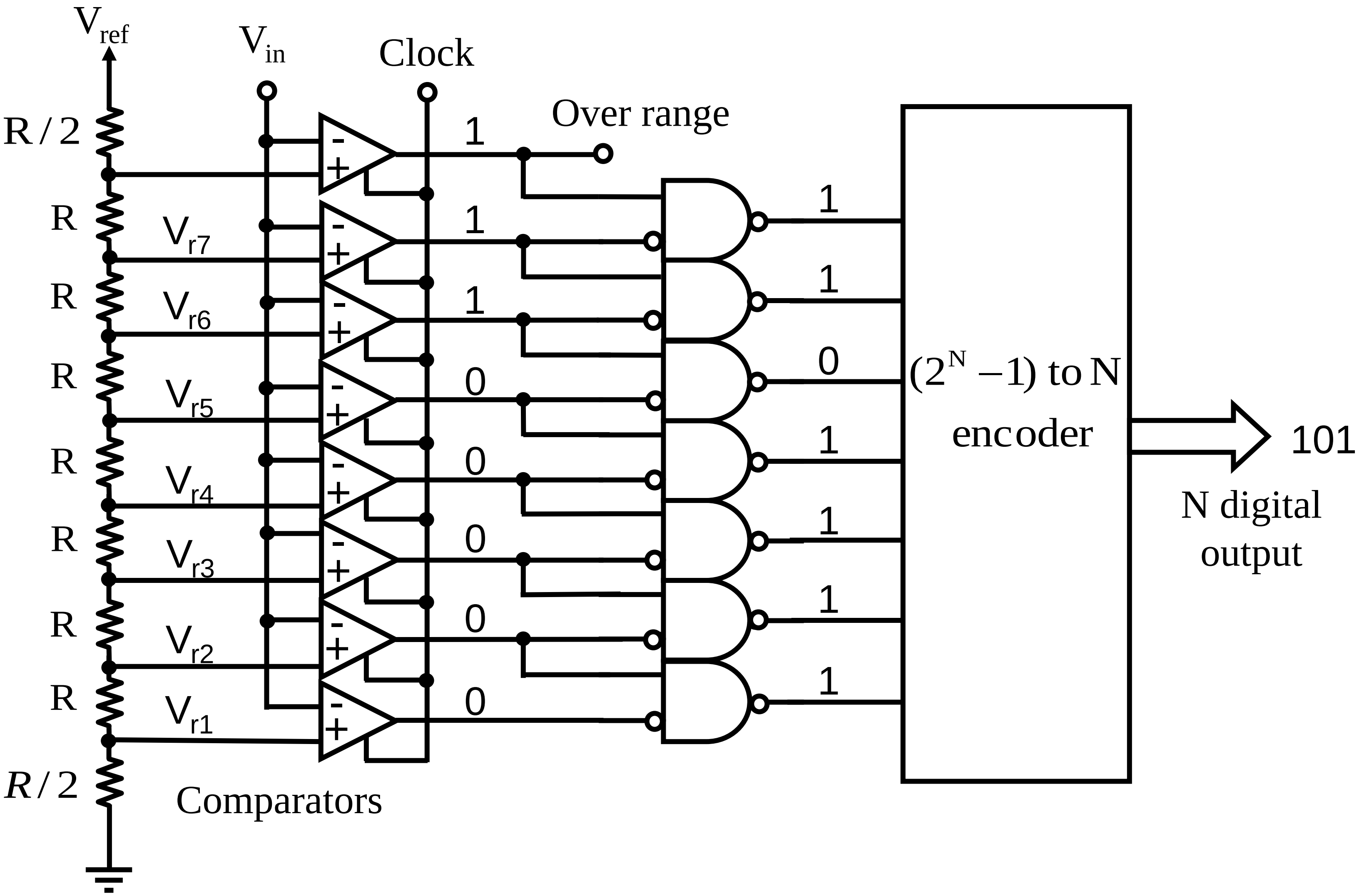
    - 30% higher than actual value
    - Circuit simulation for the ADC is required to obtain real speed

Flash(or parallel)ADC

- Very-high-speed approach, especially popular 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.
    - The NAND gate that detects a transition will have a 0 output.
    - All other NAND-gate output will be 1
  - Bubble error occurs if more than one 0 output is obtained

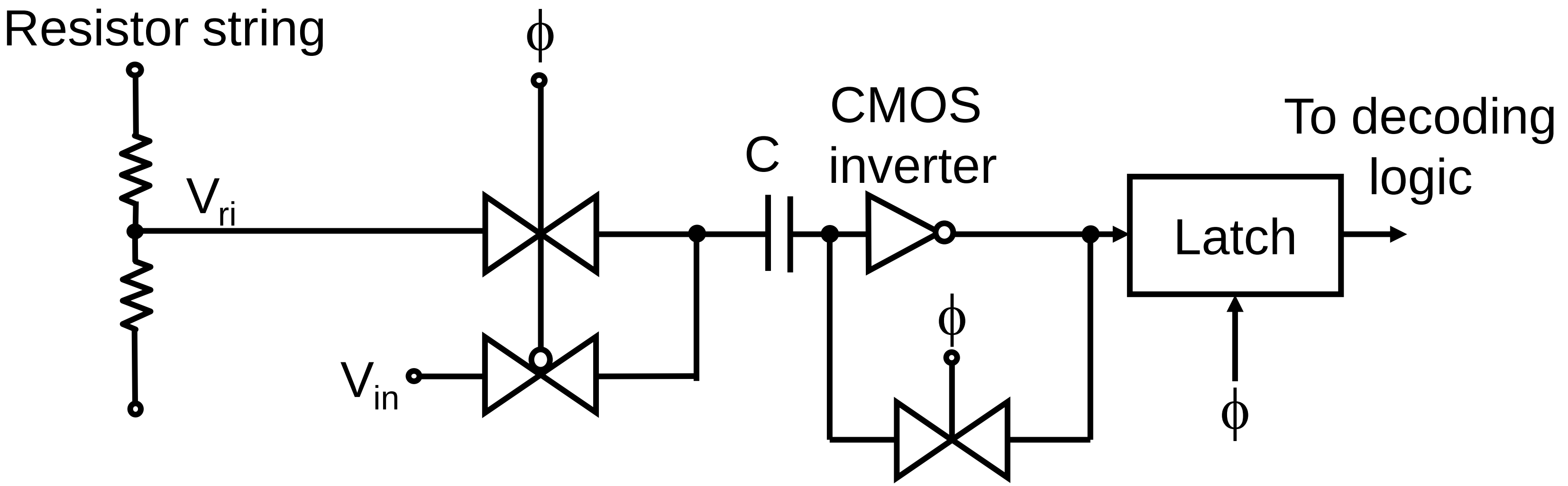
Flash(or parallel)ADC (Cont.)

- $V_{r5} < V_{in} < V_{r6}$



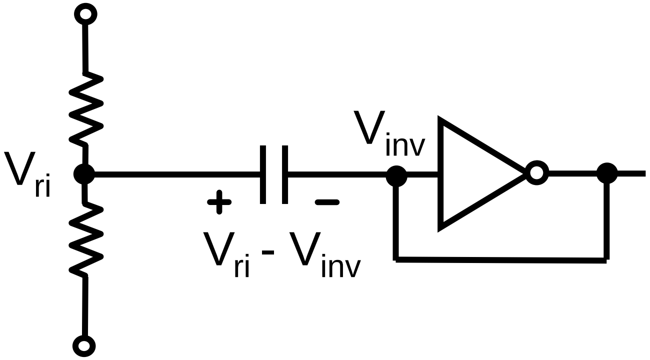
Flash(or parallel)ADC (Cont.)

- CMOS example using clocked comparator
  - It operates as a single stage OPAMP with only one pole



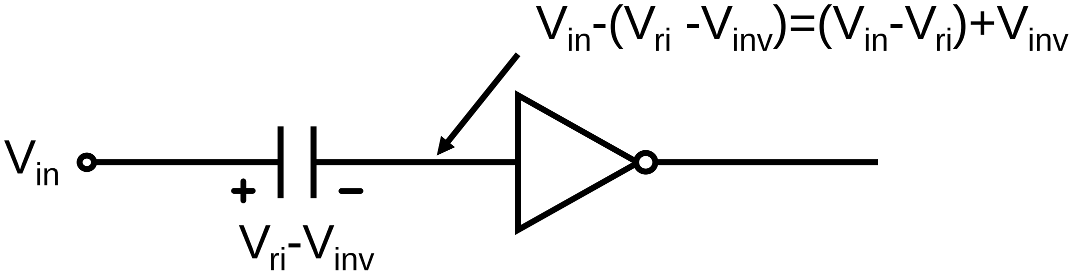
Flash(or parallel)ADC (Cont.)

- Two operation phases
  - Autozero (ϕ=1) with the inverter set to its threshold, V_{inv}, the other side of C is charge to V_{ri}



  - Signal sampling & conversion (ϕ=0)

    (V_{in}-V_{ri}) determines the polarity of the inverter output


V_{in}-(V_{ri} -V_{inv})=(V_{in}-V_{ri})+V_{inv}
V_{in}
+
−
V_{ri}-V_{inv}

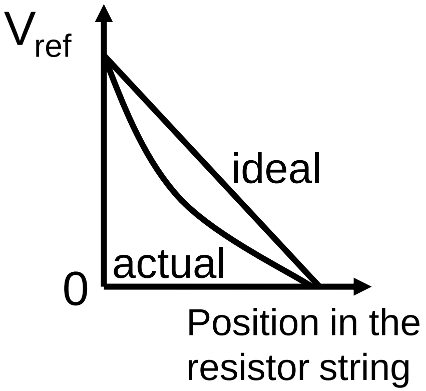
Flash(or parallel)ADC (Cont.)

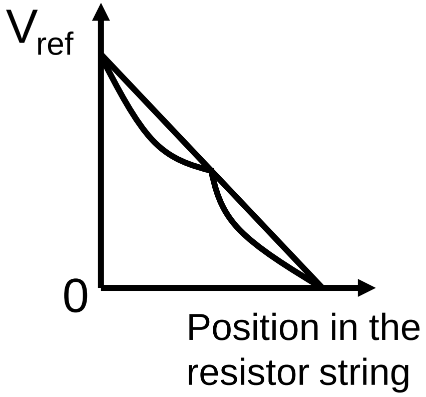
- This simple comparator suffers from poor power supply rejection. Fully differential inverter helps alleviate this shortcoming.

- 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＝25～100 for 8-bit resolution. Most often, 2 cascade inverters are used to optimize speed. Each has a gain of 5~10. Inverters are autozeroed individually, as detailedly described in chapter 7: comparators

Issues in Designing Flash ADC

- Large input capacitive load
  - Large number of comparators connected to V_{in}
  - Often limit the speed
  - Usually requires a strong and power hungry buffer to drive V_{in}
  - 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 autozero phase of clocked CMOS 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 shorted 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 500μm in metal interconnect.

    If there is clock skew between comparators greater than this, the converter will have more than 1 LSB error.
  - To reduce this error
    - Using S/H

      However, high-speed S/H can be more difficult to realize than the flash converter itself.
    - 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.

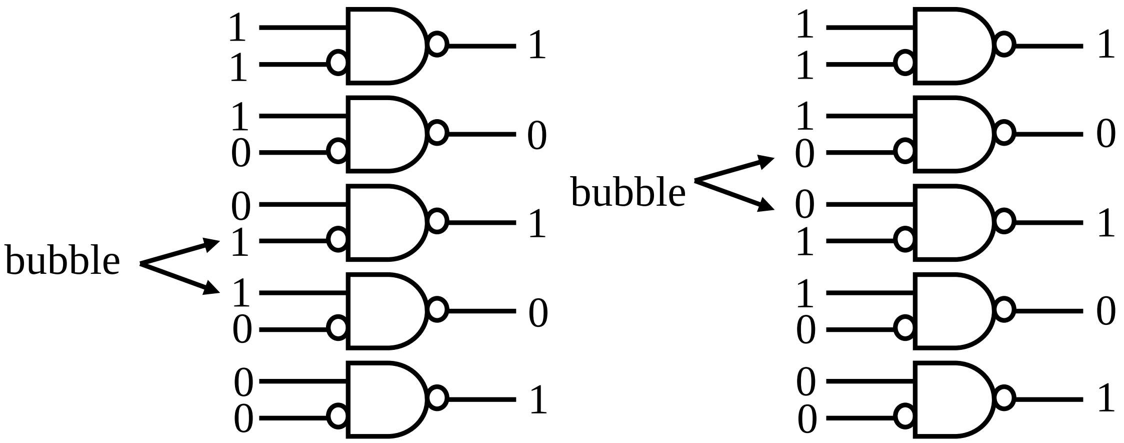
Issues in Designing Flash ADC (Cont.)

- 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 hundreds of MHz, it is difficult to keep power-supply noise below a few tens of a milivolt.
  - To reduce this effect
    - Running differential clocks closely together will help prevent the signals from being coupled into the substrate or through the air.
    - 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.
    - On-chip power-supply bypassing is a necessity, and it is also necessary to make sure the power-supply bypassing circuitry doesn’t form a resonant circuit with the bonding wire.

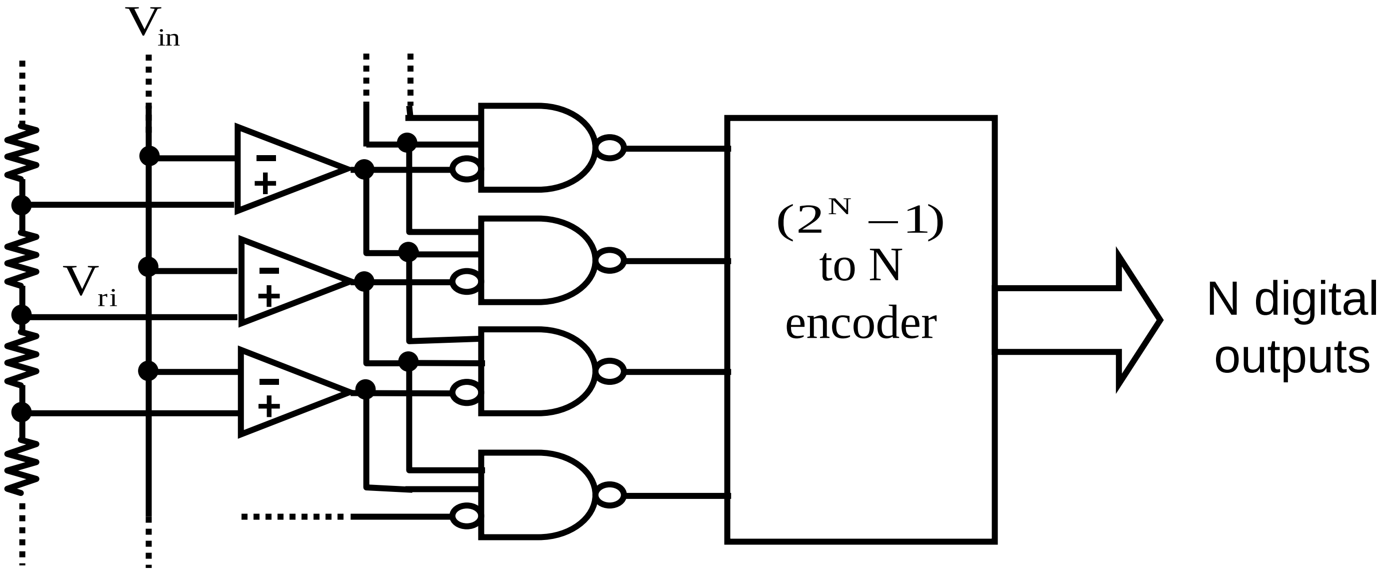
Issues in Designing Flash ADC (Cont.)

- 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-input 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.676~p.677 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 of 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.

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.

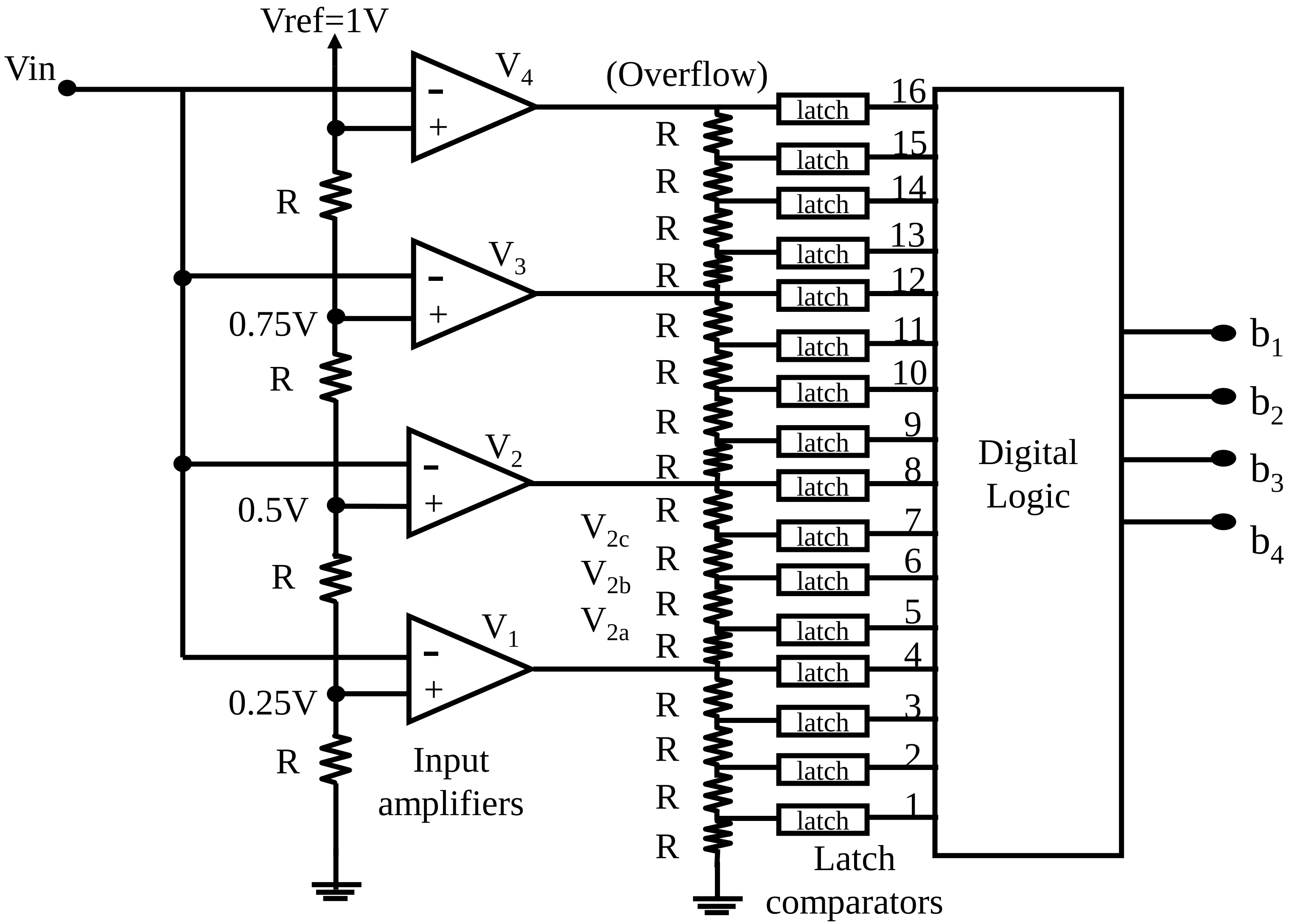
Interpolating ADC (Cont.)

- Example: 4-bit
  - The input amplifiers have a maximum gain of 10
    - logic level = 0V, 5V

      latch threshold ≈ 2.5V

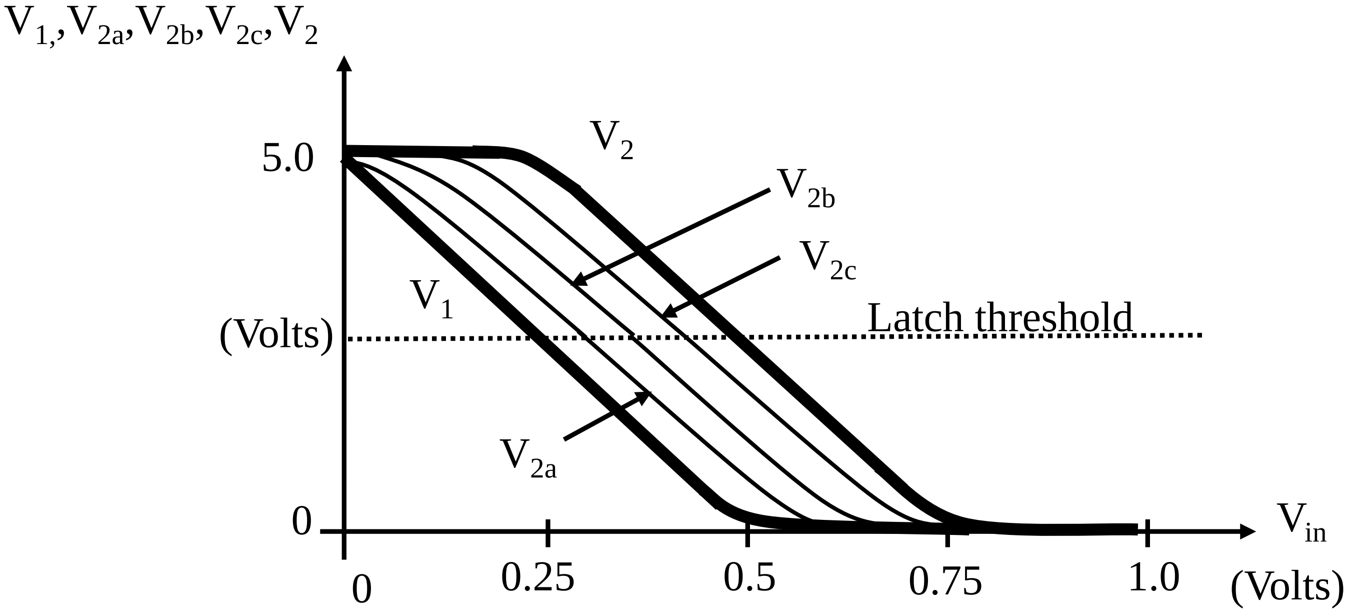
      voltage difference between adjacent nodes of resistor-string = 0.25V
    - $\left|\text{gain}\right| \le \dfrac{2.5\text{V}}{0.25\text{V}} = 10$
  - For good linearity, the interpolated signals need only cross the latch threshold at correct points, while the rest of the interpolated signals response are of secondary importance.

Interpolating ADC (Cont.)



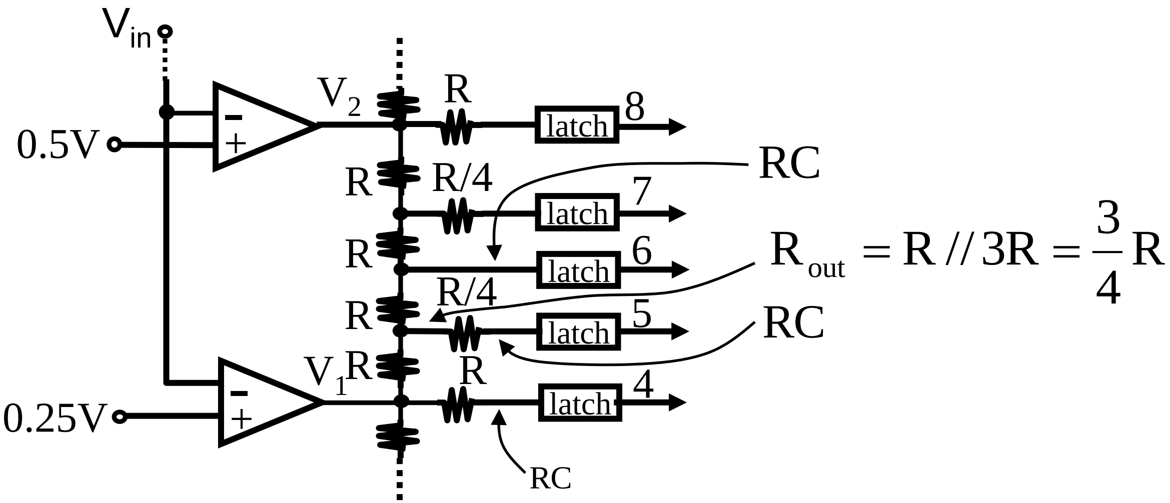
Interpolating ADC (Cont.)

- 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.

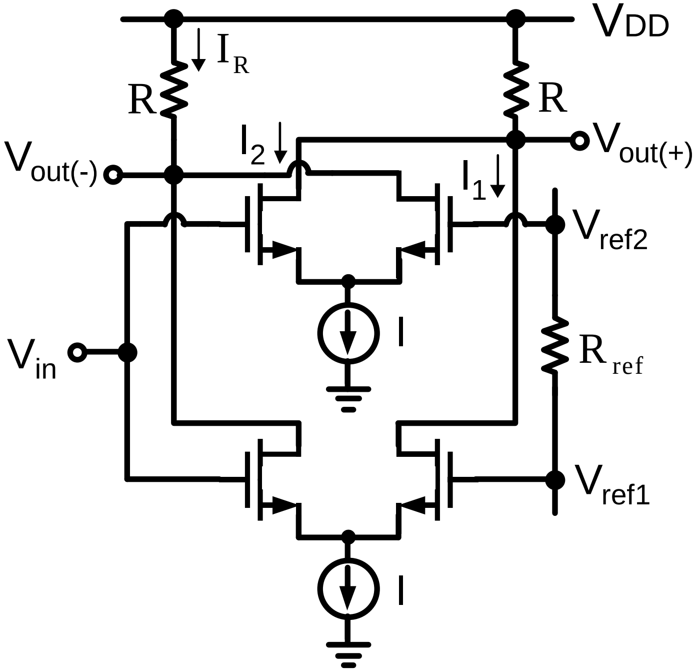


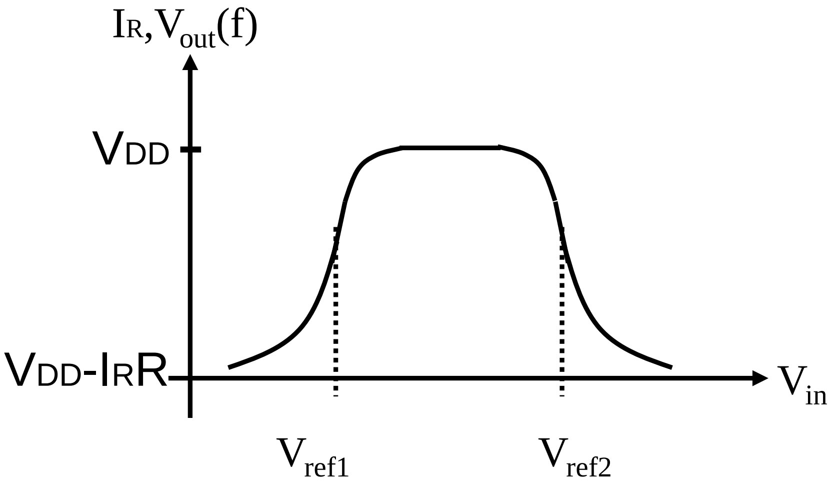
$$R_{out} = R // 3R = \frac{3}{4} R$$

- Other implementation methods
  - Interpolating using current mirrors or capacitors

Folding ADC

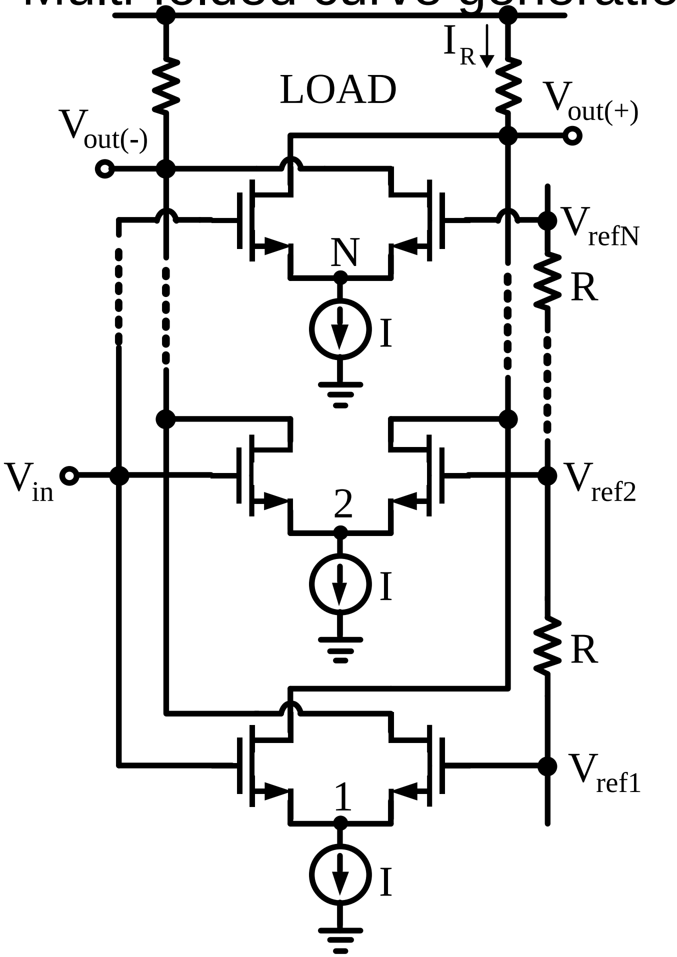
- Heavy input load (similar to flash and heavier than interpolating)
- Reduced number of latch comparators(compared to flash and interpolating)
- Example1
  - Two-folded curve generation

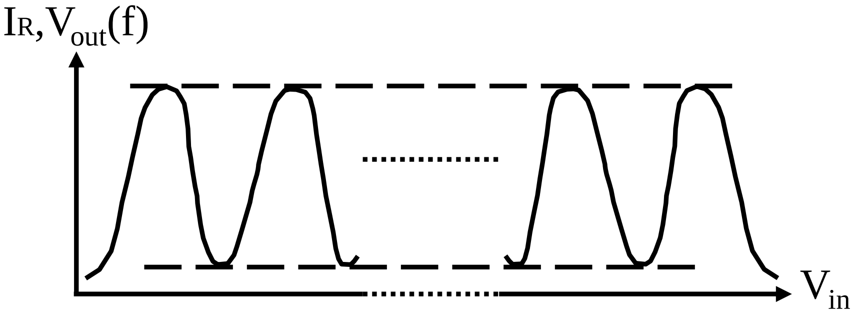




Folding ADC (Cont.)

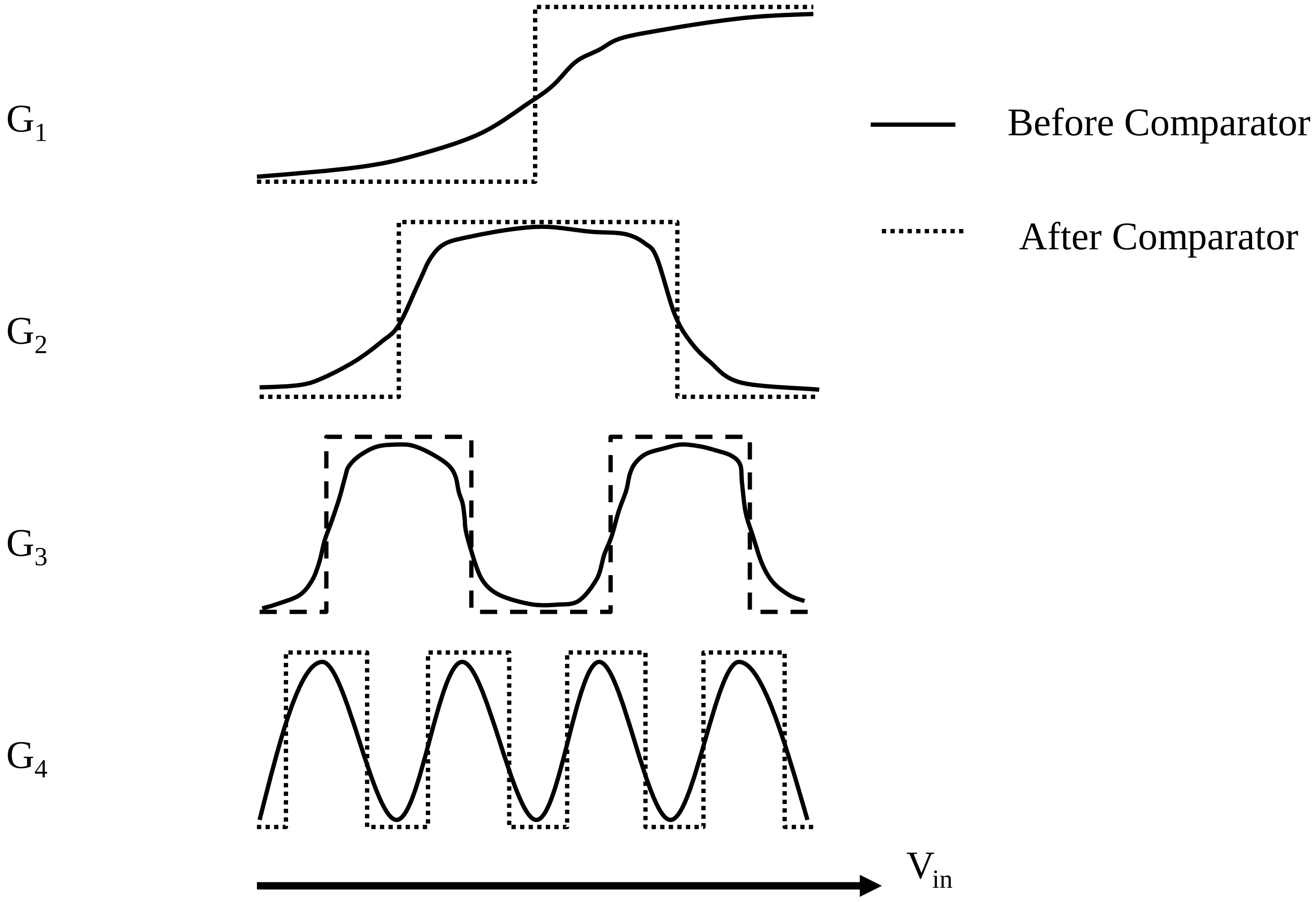
- Multi-folded curve generation





Folding ADC (Cont.)

- Gray coded curves



Folding ADC (Cont.)

- Gray to binary converter

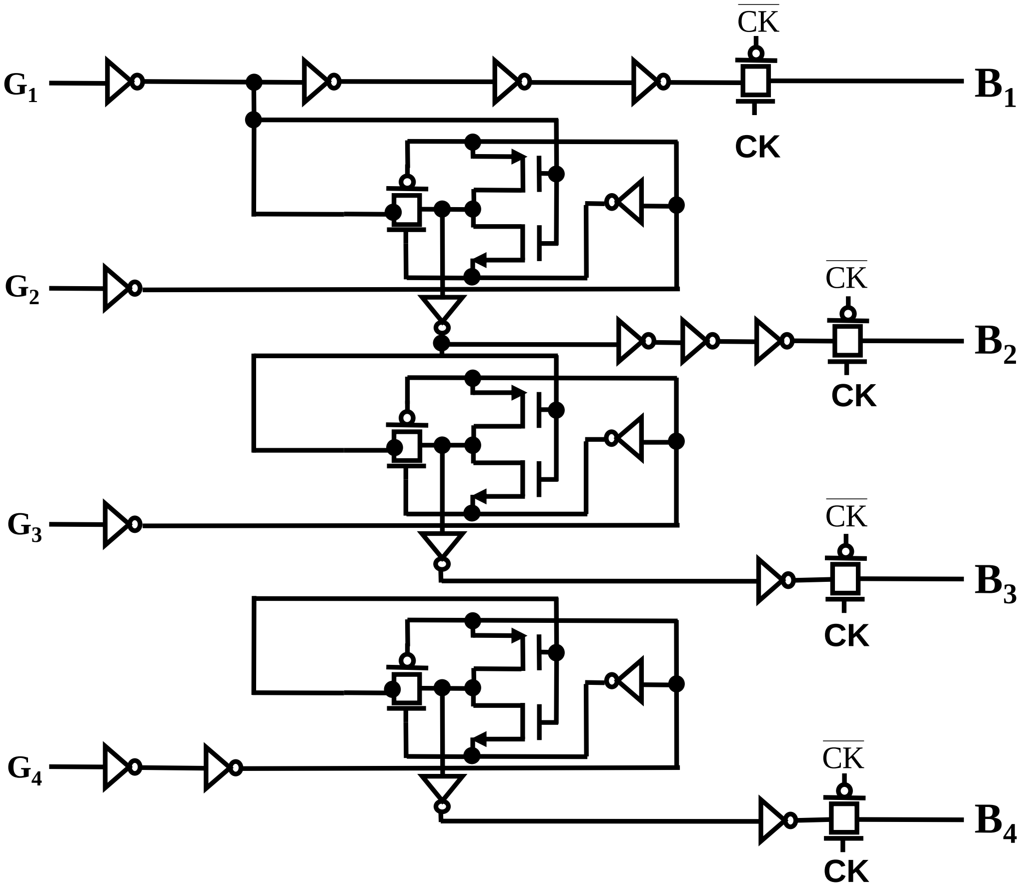
  The relation between a Gray code and a binary code

$B_1 = G_1$

$B_2 = G_2 \otimes B_1$

....

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



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

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.

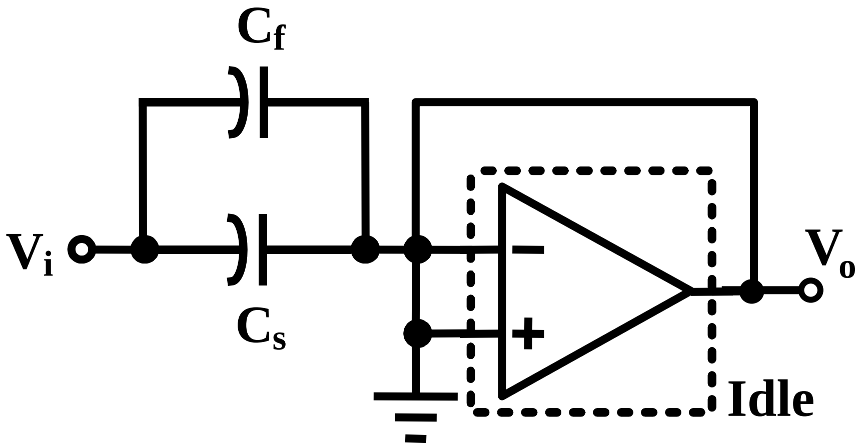
    Frequency of folding curve = Input frequency x Folding rate

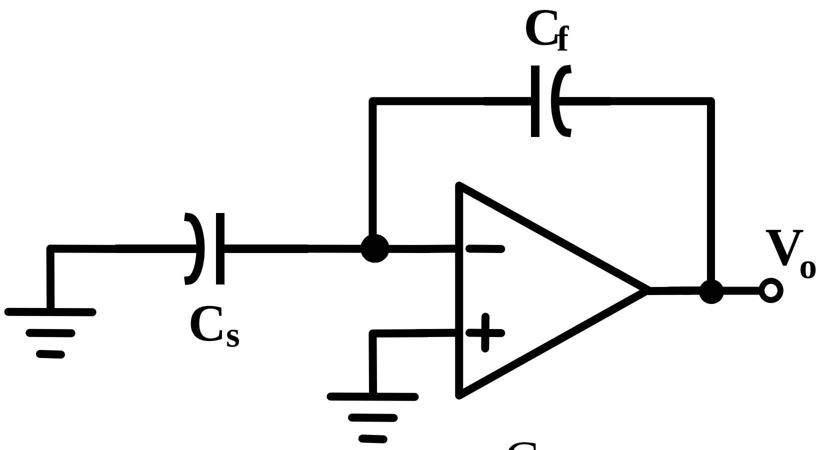
    This multiplying effect limits the practical folding rate used in high frequency converters.
  - Differential circuits are almost always used in practical implementation.

Multiplying DAC (MDAC)

- Fully-differential circuits are normally used
  (For simplicity, a single-ended circuit is used here)

- Operational principle (two phases)
  - ◆ Sampling phase : sample the V_i with C_s and C_f



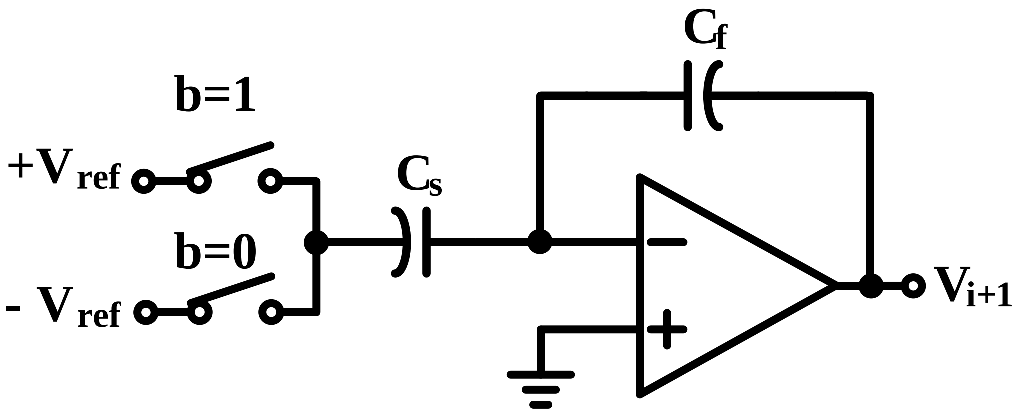


$$V_o = (1+\frac{C_S}{C_f})V_i$$

$$\text{If } C_S = C_f \Rightarrow V_o = 2V_i$$

Multiplying DAC (MDAC) (Cont.)

- Amplify phase :

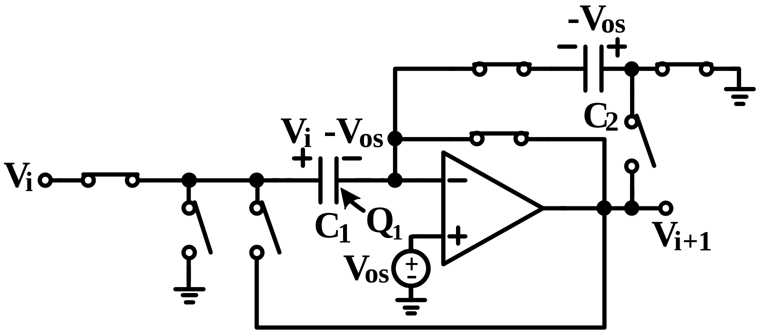


$$V_{i+1}=\begin{cases}\left(1+\frac{C_s}{C_f}\right)\cdot V_i+V_{ref}\left(\frac{C_s}{C_f}\right);b=0\\ \left(1+\frac{C_s}{C_f}\right)\cdot V_i-V_{ref}\left(\frac{C_s}{C_f}\right);b=1\end{cases}\xrightarrow{C_s=C_f}V_{i+1}=\begin{cases}2\cdot\left(V_i-\left(-\frac{1}{2}V_{ref}\right)\right);b=0\\ 2\cdot\left(V_i-\left(+\frac{1}{2}V_{ref}\right)\right);b=1\end{cases}$$

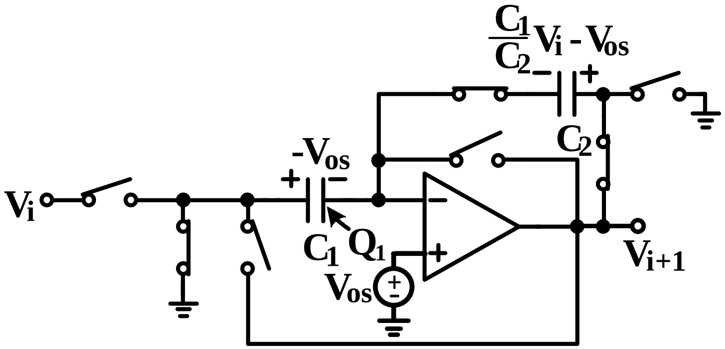
➢ The accuracy of gain depends on capacitor matching

Capacitor Ratio-Independent Multiplication

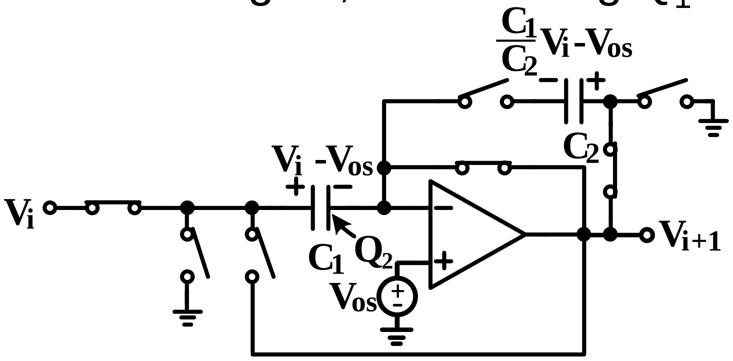
- Accurate multiply-by-two gain amplifier:
  - Does not rely on any capacitor matching
  - Four clock cycles are required
- Operational principle (four phases)
  - Phase 1: Sample remainder and cancel input-offset voltage



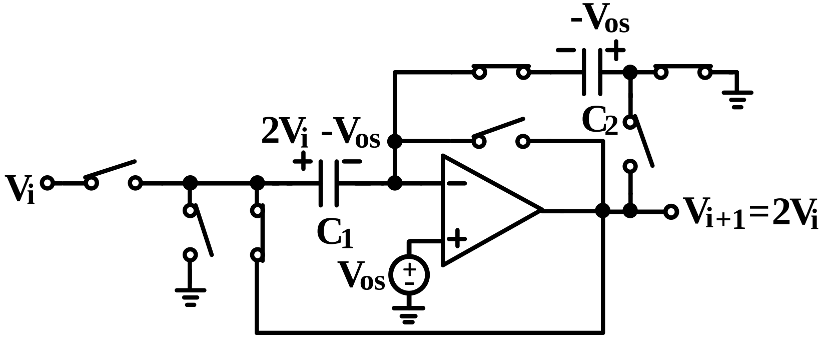
  - Phase 2: Transfer charge Q_1 from C_1 to C_2



  - Phase 3: Sample input signal with C_1 again, after storing Q_1 on C_2



  - Phase 4: Combine Q_1 and Q_2 on C_1, and connect C_1 to output

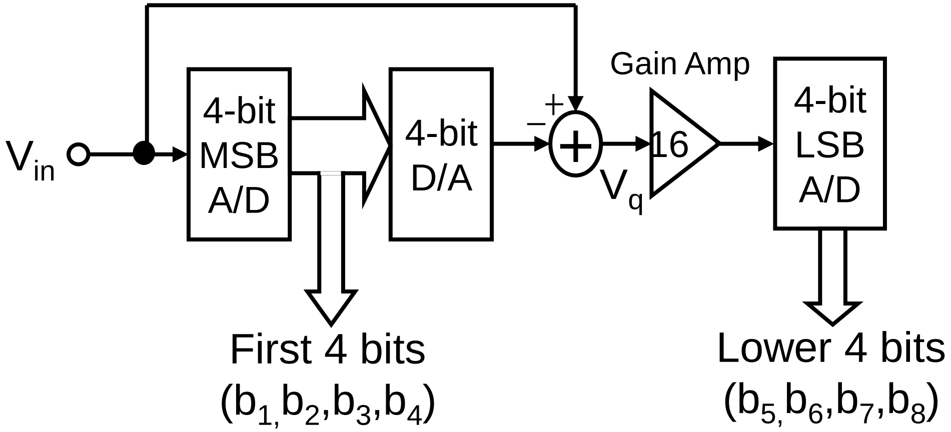


Two-Step (or Subranging) ADC

- Compared to flash ADC
  - Less area
  - Less power
  - Less input capacitive loading
  - The voltages of comparators need to resolve are less stringent
  - Larger latency
  - Can’t realize very high speed due to the use of S/H

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

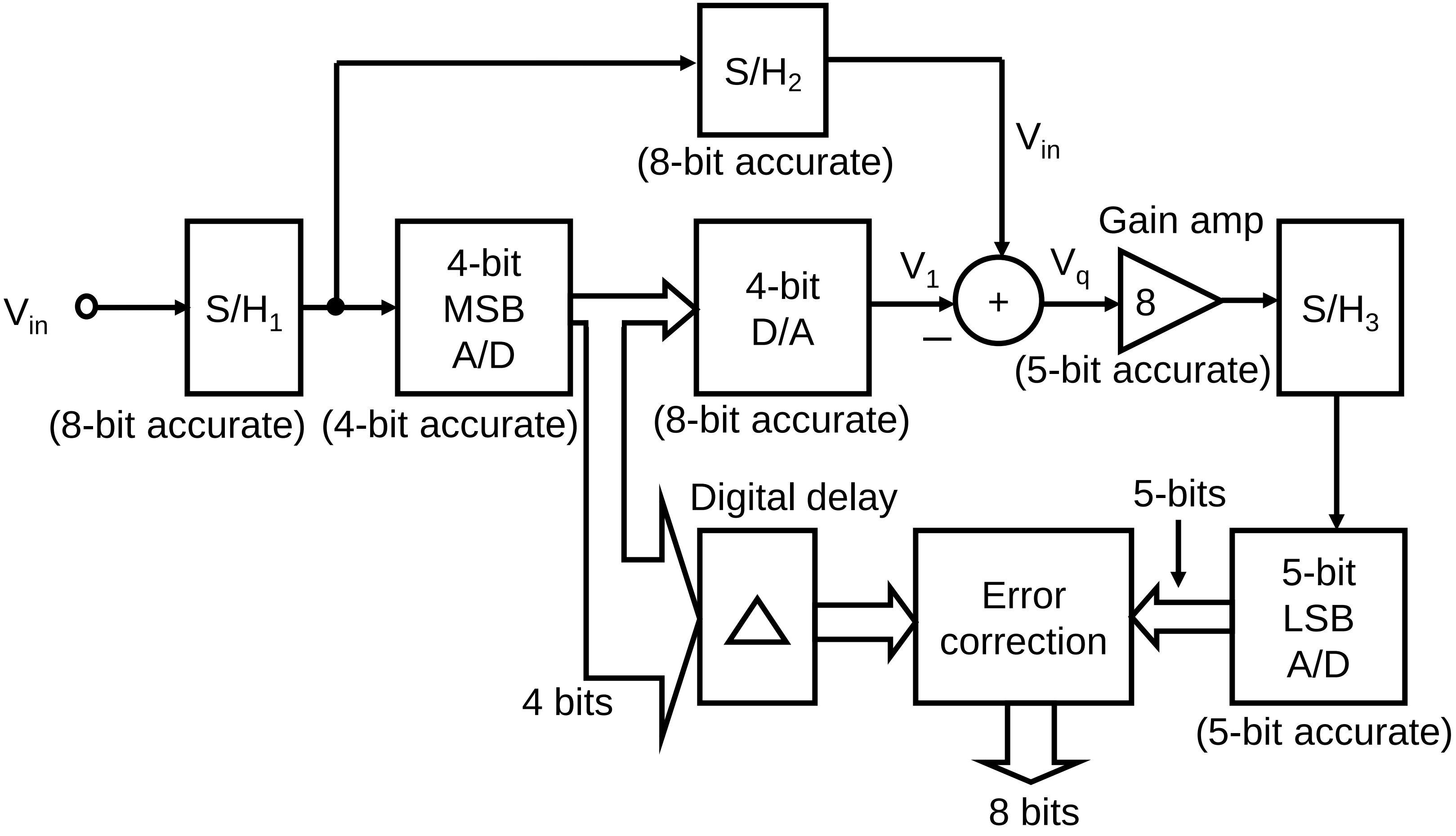
- 8-bit example



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

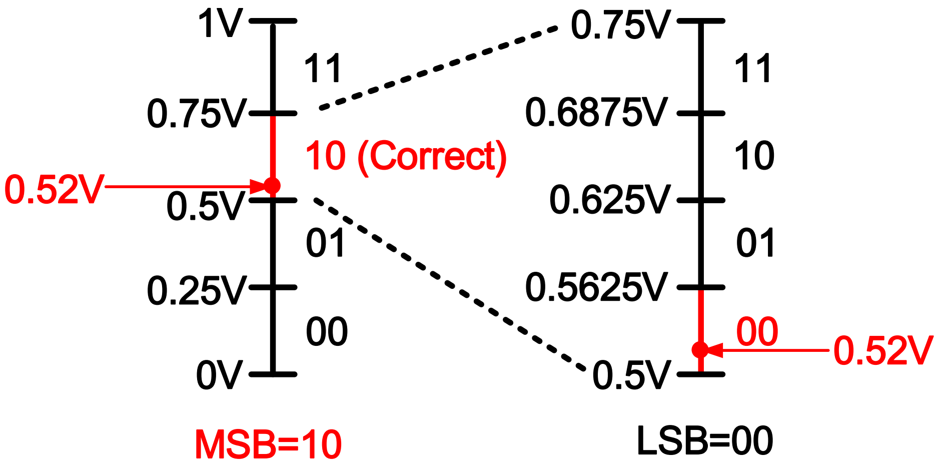


Two-Step (or Subranging) ADC (Cont.)

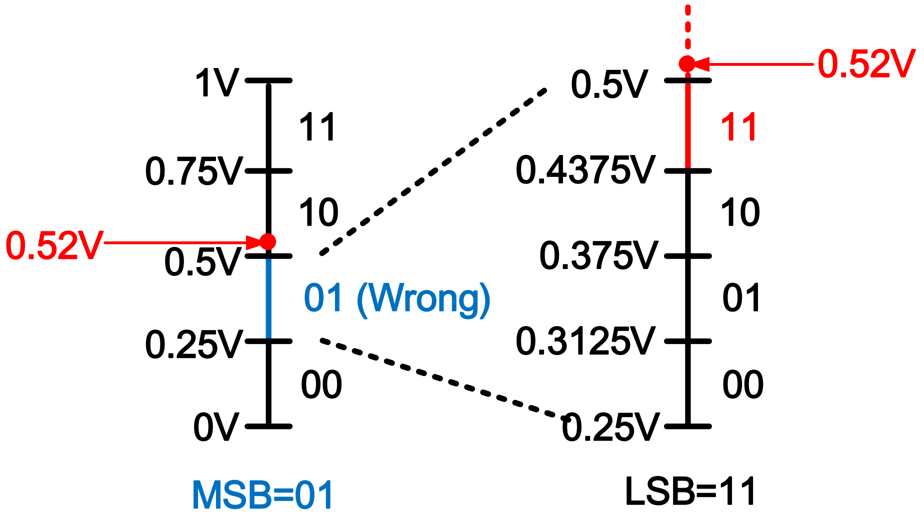
- Signal is pipelined
  - Need more S/H to maintain high speed
  - Speed is halved if only one S/H is used
- With error correction, relaxed circuit accuracy of internal ADCs and gain stage
- Accuracy required for each block is shown in the figure above (The reasons are included in p.679~p.680 of the textbook)

Error Correction Example of Two-Step ADC

- 2-bit MSB + 2-bit LSB
  - MSB is resolved correctly
    - MSB+LSB → 1000
    - Correct

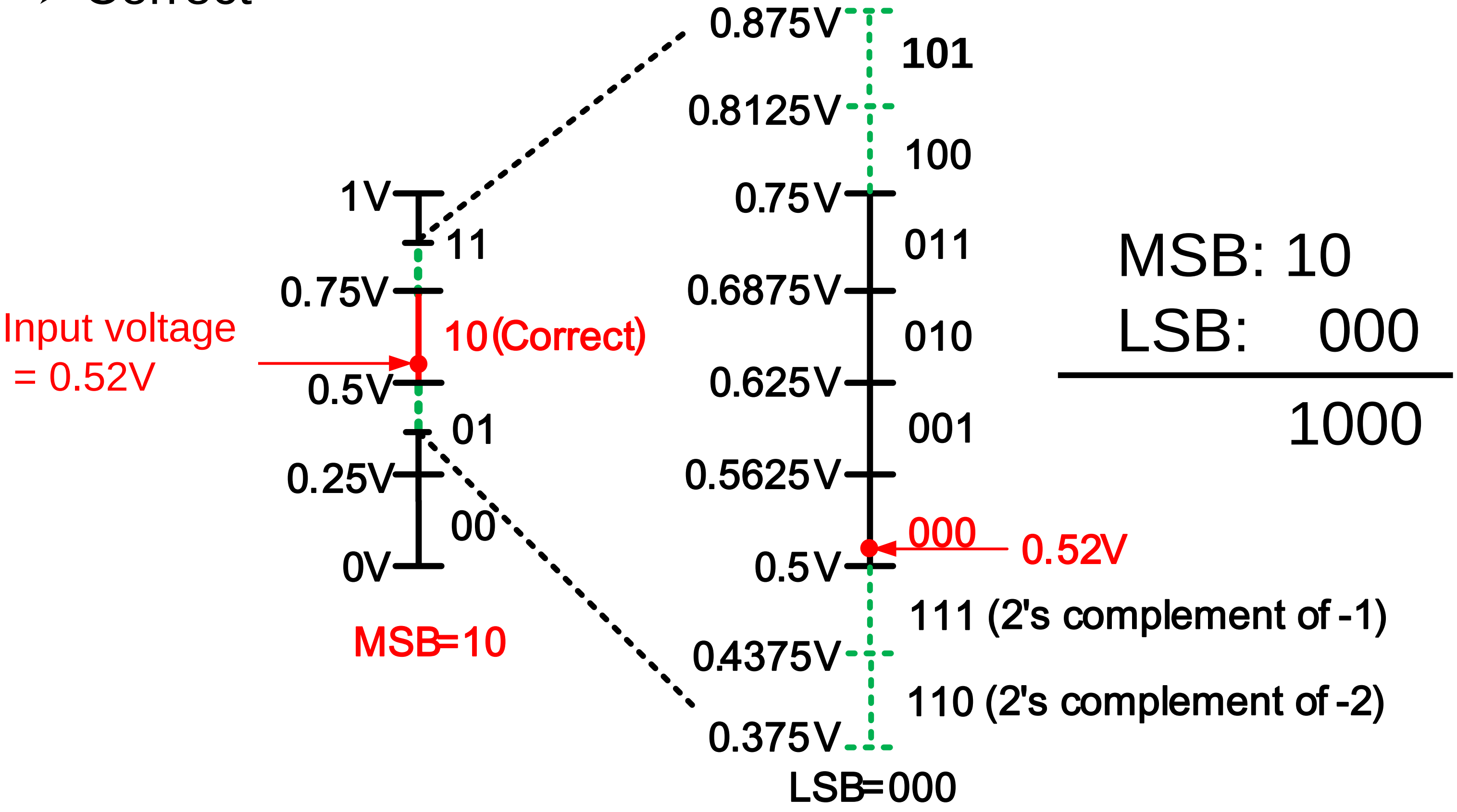


  - MSB is resolved wrongly
    - MSB+LSB → 0111
    - Wrong



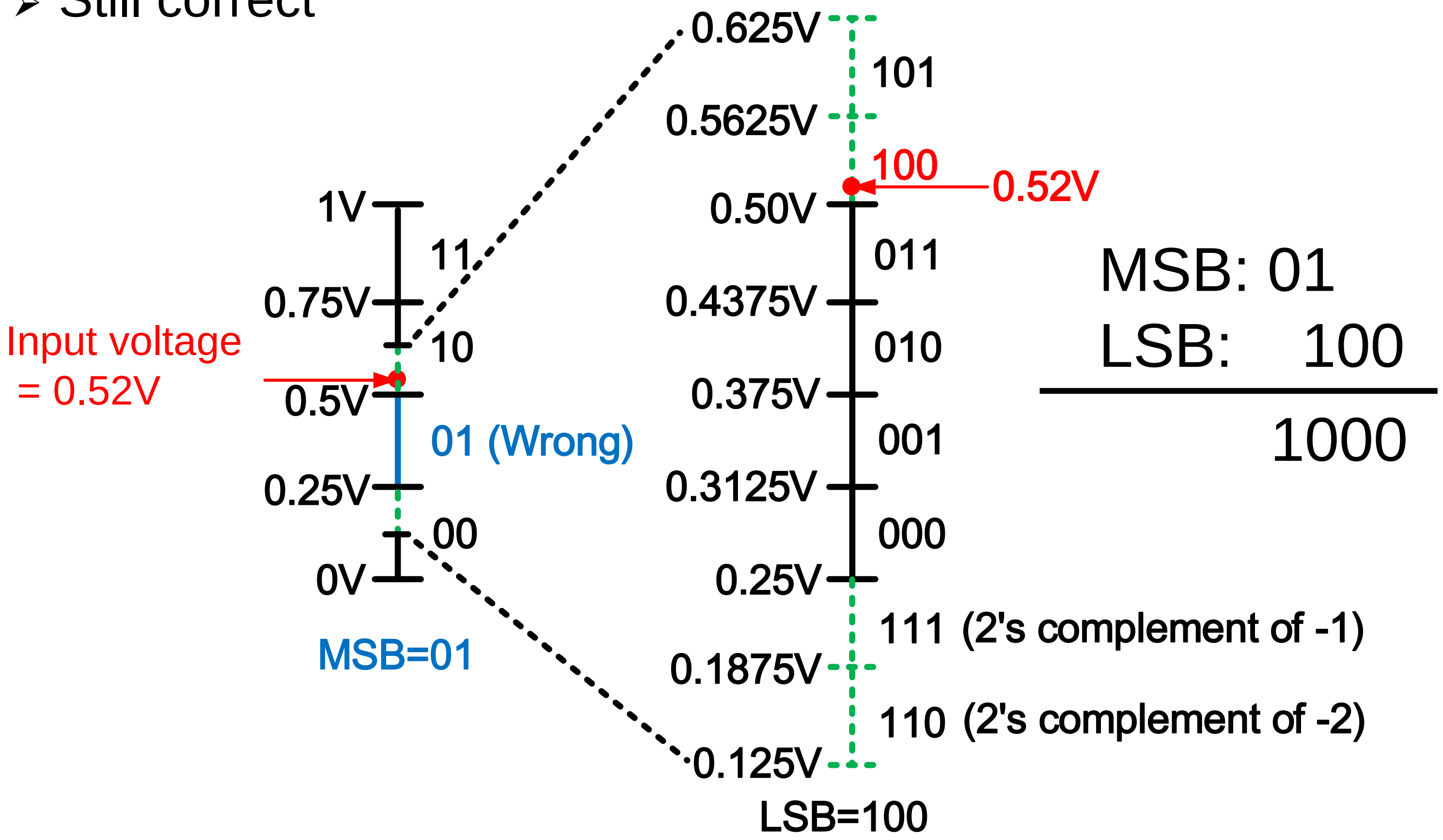
Error Correction Example of Two-Step ADC (Cont.)

- 2-bit MSB + 3-bit LSB (including 1-bit error correction)
  - ◆ MSB is resolved correctly
    - ➢ MSB+LSB → 1000
    - ➢ Correct



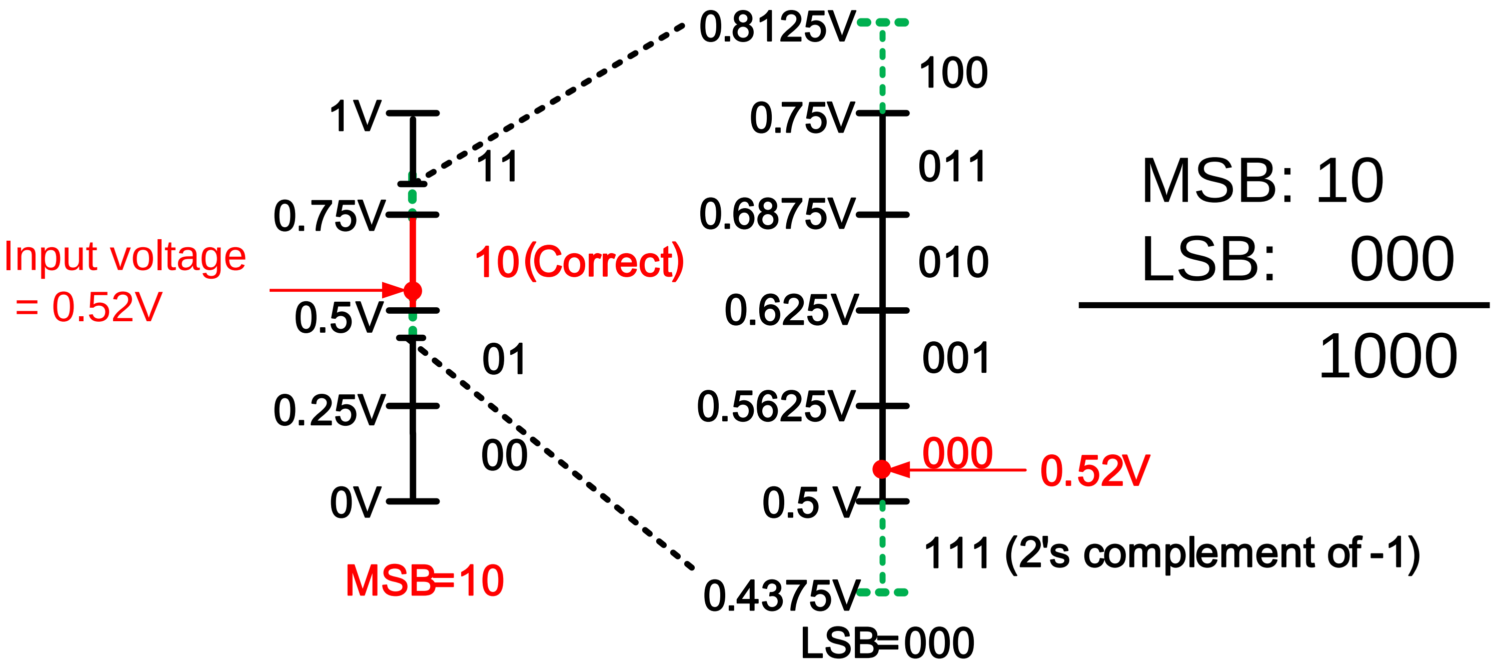
Error Correction Example of Two-Step ADC (Cont.)

- 2-bit MSB + 3-bit LSB (including 1-bit error correction)
  - MSB is resolved wrongly
    - MSB+LSB → 1000
    - Still correct



Error Correction Example of Two-Step ADC (Cont.)

- 2-bit MSB + 2.5-bit LSB (including 0.5-bit error correction)
  - ◆ MSB is resolved correctly
    - ➢ MSB+LSB → 1000
    - ➢ Correct

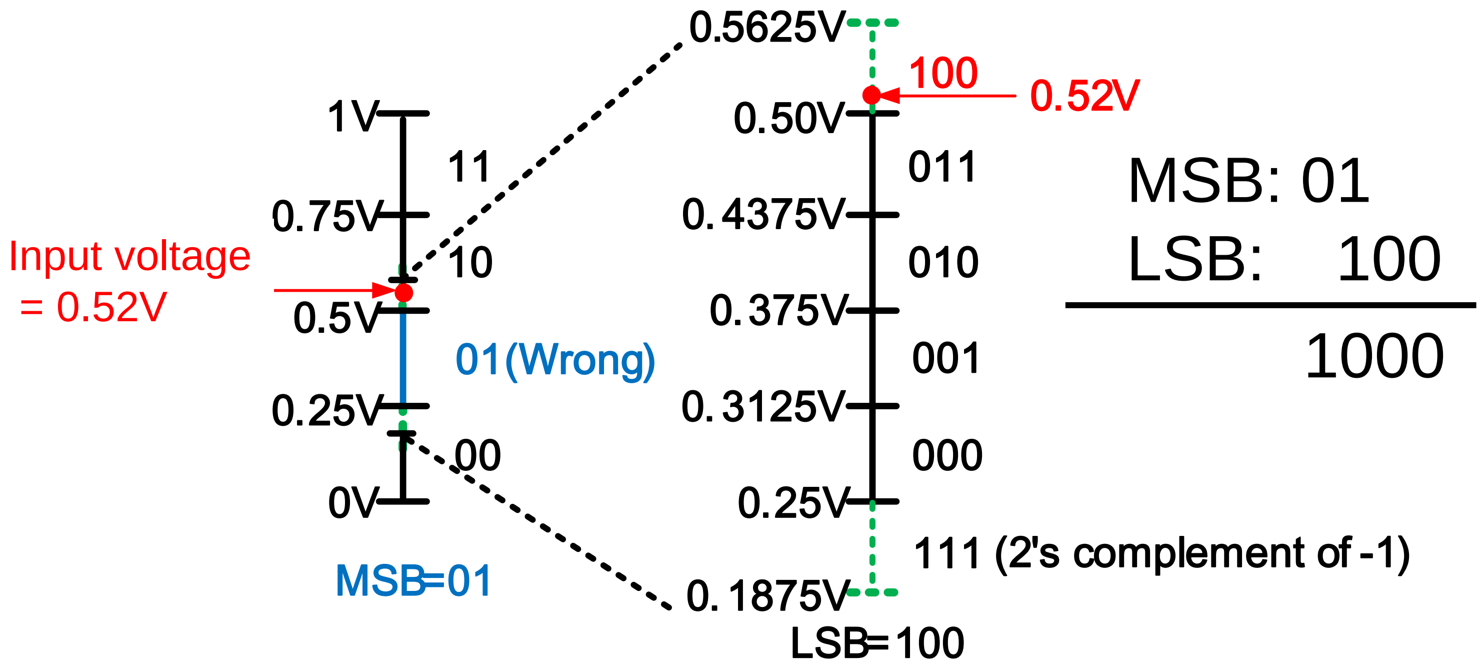


MSB: 10
LSB: 000

1000

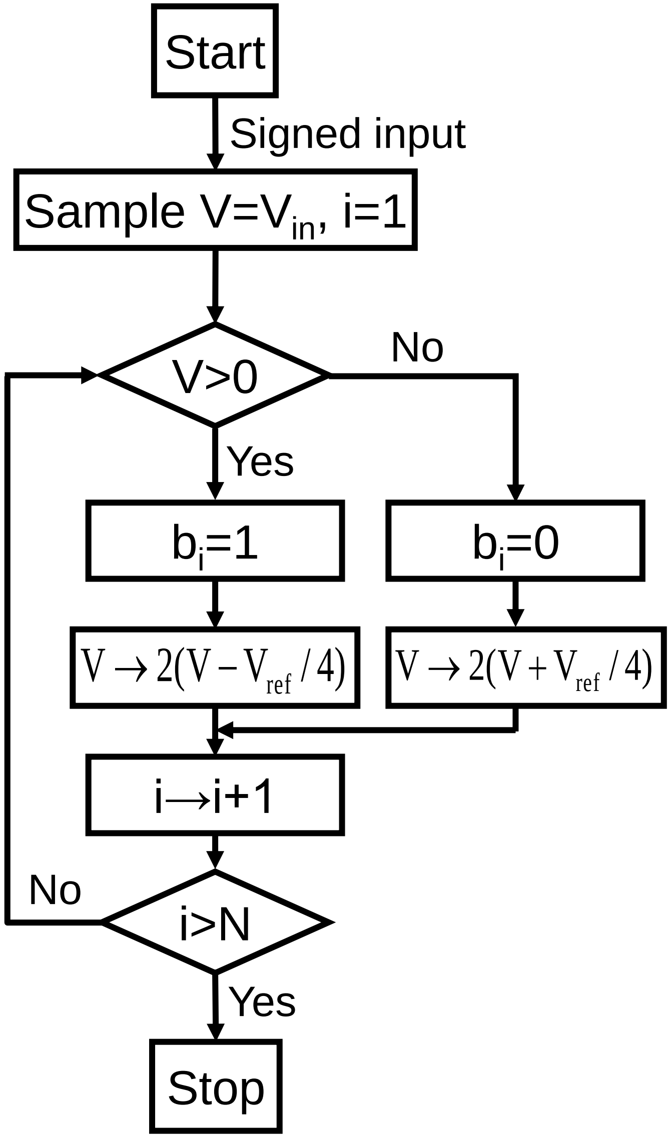
Error Correction Example of Two-Step ADC (Cont.)

- 2-bit MSB + 2.5-bit LSB (including 0.5-bit error correction)
  - MSB is resolved wrongly
    - MSB+LSB → 1000
    - Still correct



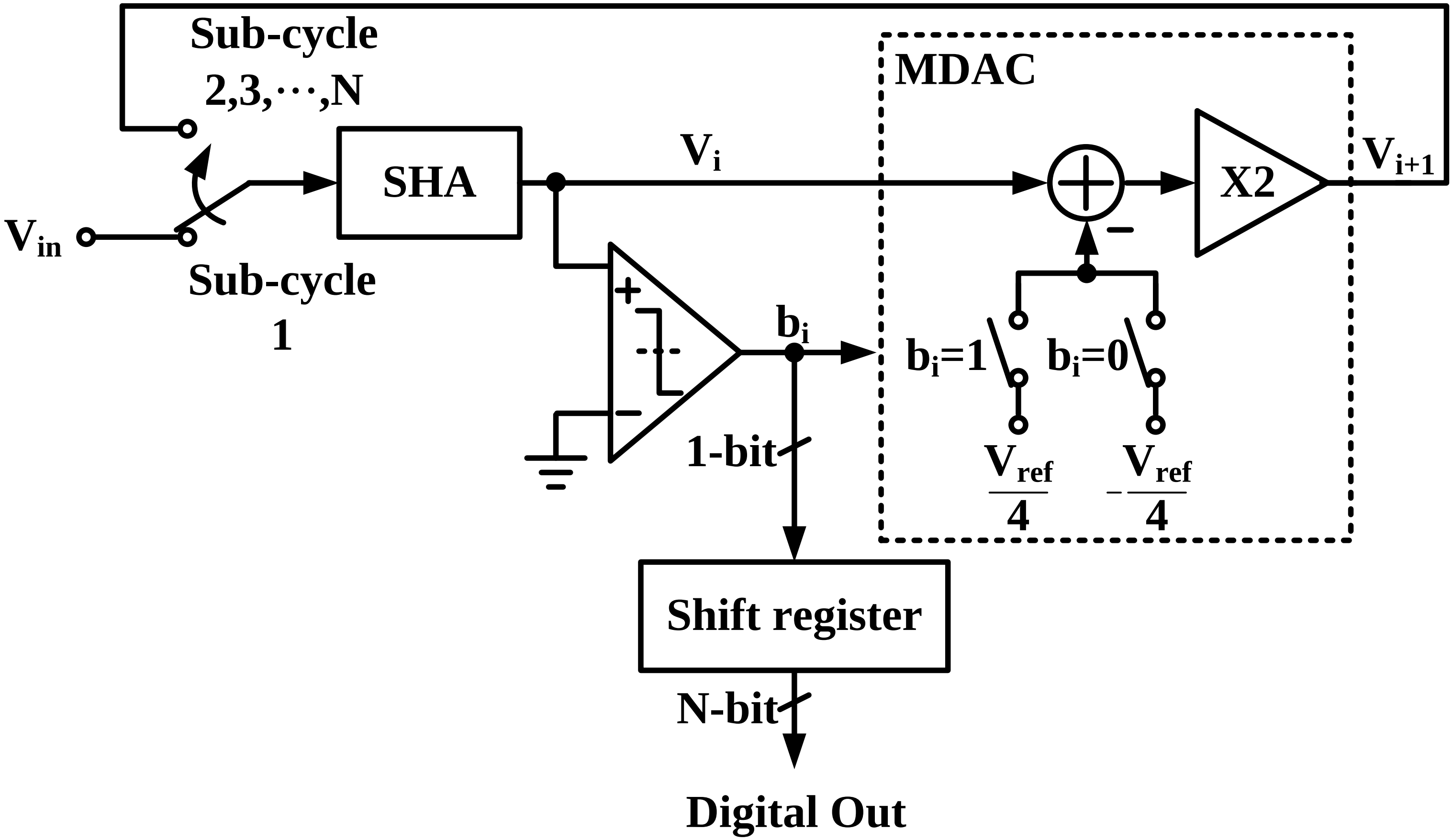
Algorithmic(or cyclic)ADCs

- A successive approximation converter halves the reference voltage in each cycle, an algorithm converter doubles the error voltage while leaving the reference voltage unchanged
- 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
  - ◆ Sample-and-hold amplifier (SHA)
  - ◆ Multiplying DAC (MDAC)
    - ➢ Two clock cycles
    - ➢ Rely on capacitor matching



Algorithmic(or cyclic)ADCs (Cont.)

- 1-bit/cycle
- Input (V_{in}) range : $-\frac{1}{2}\mathrm{V}_{\mathrm{ref}} \sim +\frac{1}{2}\mathrm{V}_{\mathrm{ref}}$



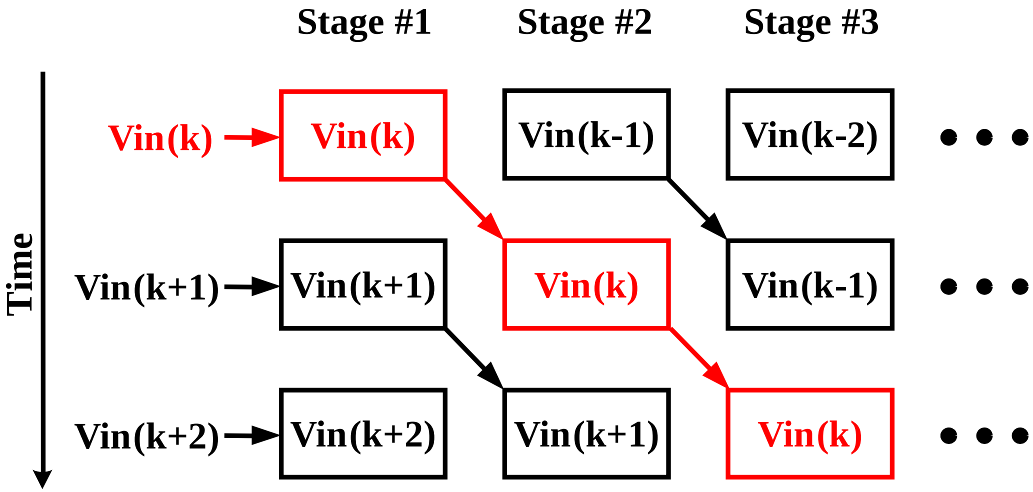
Pipelined ADC

- 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, 2.5-bit/stage – with digital error correction
  In general, 1.5-bit/stage is the optimum with respect to speed, area and power.

- Current state-of-art is 12 to 16 bits for pipelined ADC with digital error correction at hundreds of MHz.

Pipelined ADC (Cont.)

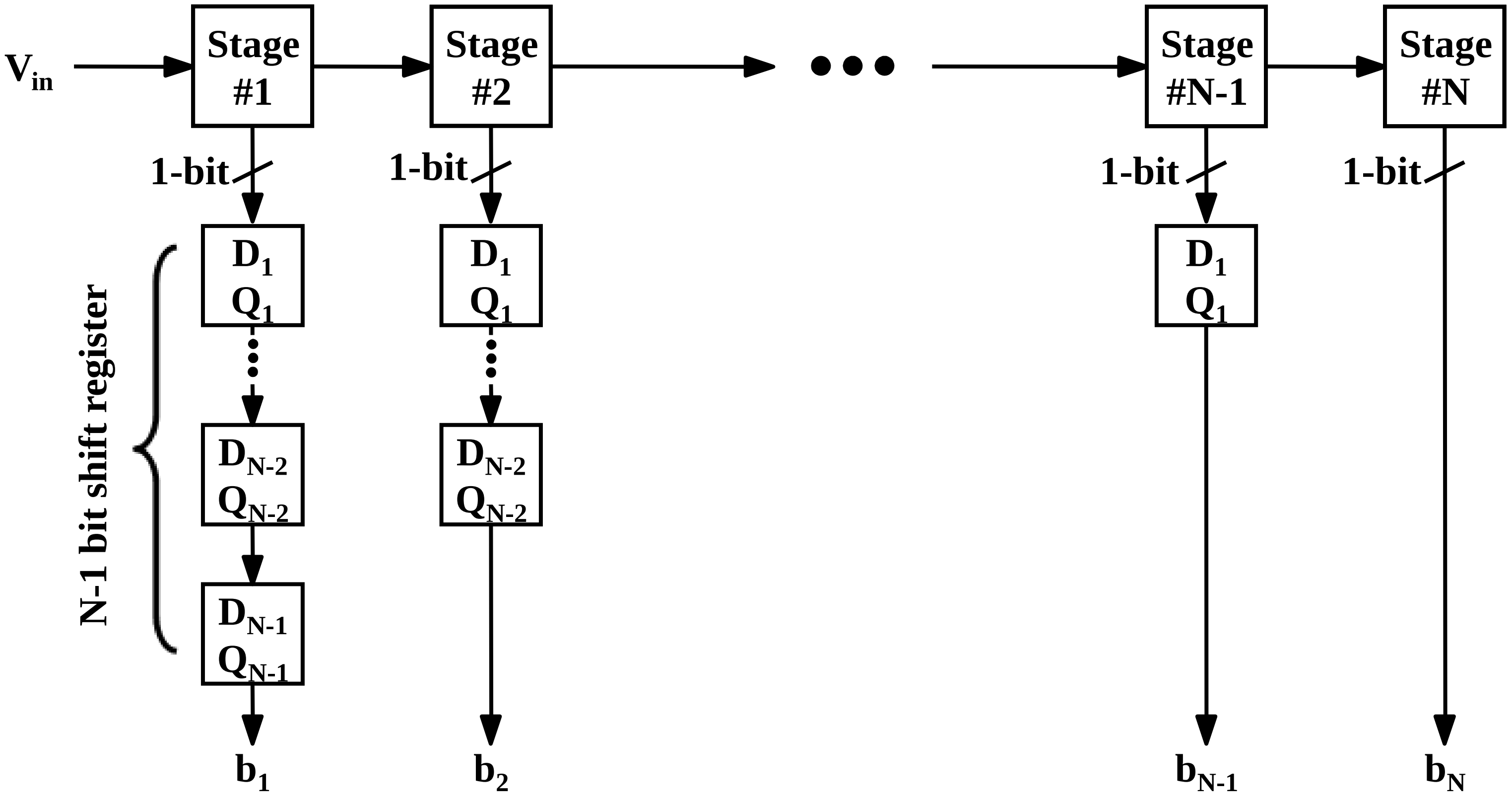
- Once the stage #1 completes its work, it does not sit idle while the remaining lower bits are found, but immediately starts work on the next input sample.



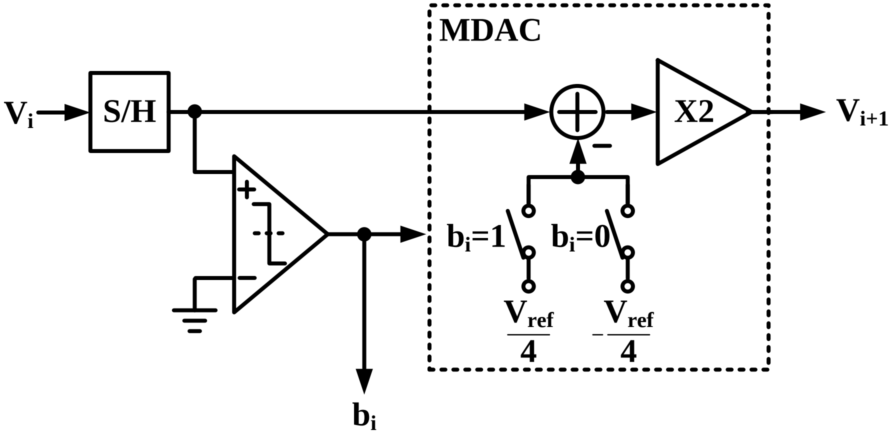
- Conversion rate equals clock rate
- It takes N clock cycles for each input signal (latency is N)
- Circuit complexity is proportional to N
- Small area

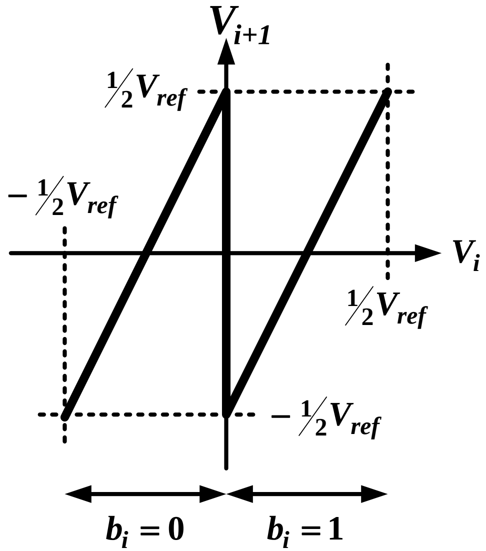
1-Bit/Stage Pipelined ADC

- Block diagram (no digital error correction)



1-Bit/Stage Pipelined ADC (Cont.)





(Operational principle of MDAC, p.12-47~p.12-48)

- Each stage contains an S/H to store the input signal. This S/H allows the proceeding stage to be immediately used to process its next input signal before the succeeding stage has finished.
- For a signed conversion, the input is compared to 0 V.

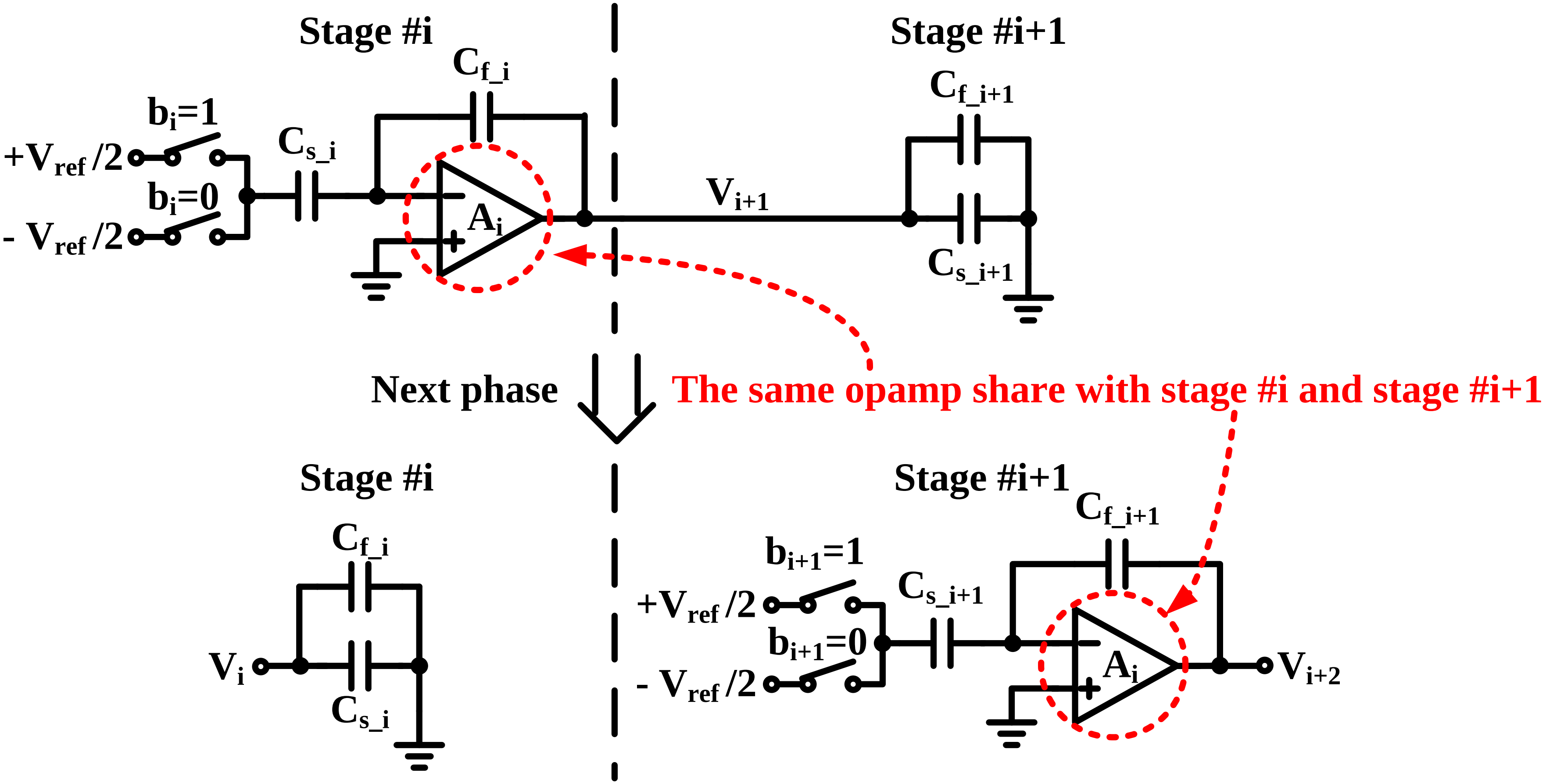
  If $V_{in} > 0$, $V_{out} = 2V_{in} - (V_{ref}/2) = 2(V_{in} - (V_{ref}/4))$ and $B_{out} = 1$.

  Otherwise, $V_{out} = 2V_{in} + (V_{ref}/2)$, and $B_{out} = 0$
- The i^th S/H can be incorporated into the (i-1)^th MDAC except for the first stage of the pipelined ADC

1-Bit/Stage Pipelined ADC (Cont.)

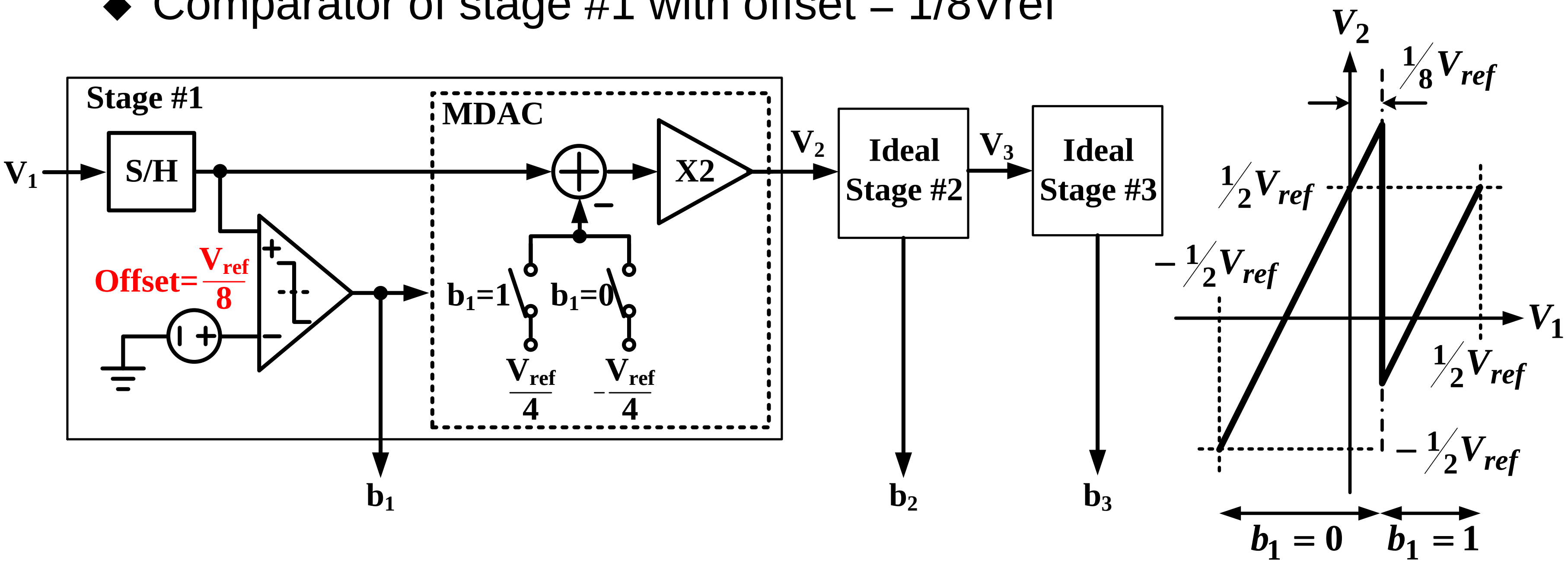
- Opamp sharing → Power and area saving

  Opamps are shared between two consecutive stages



1-Bit/Stage Pipelined ADC (Cont.)

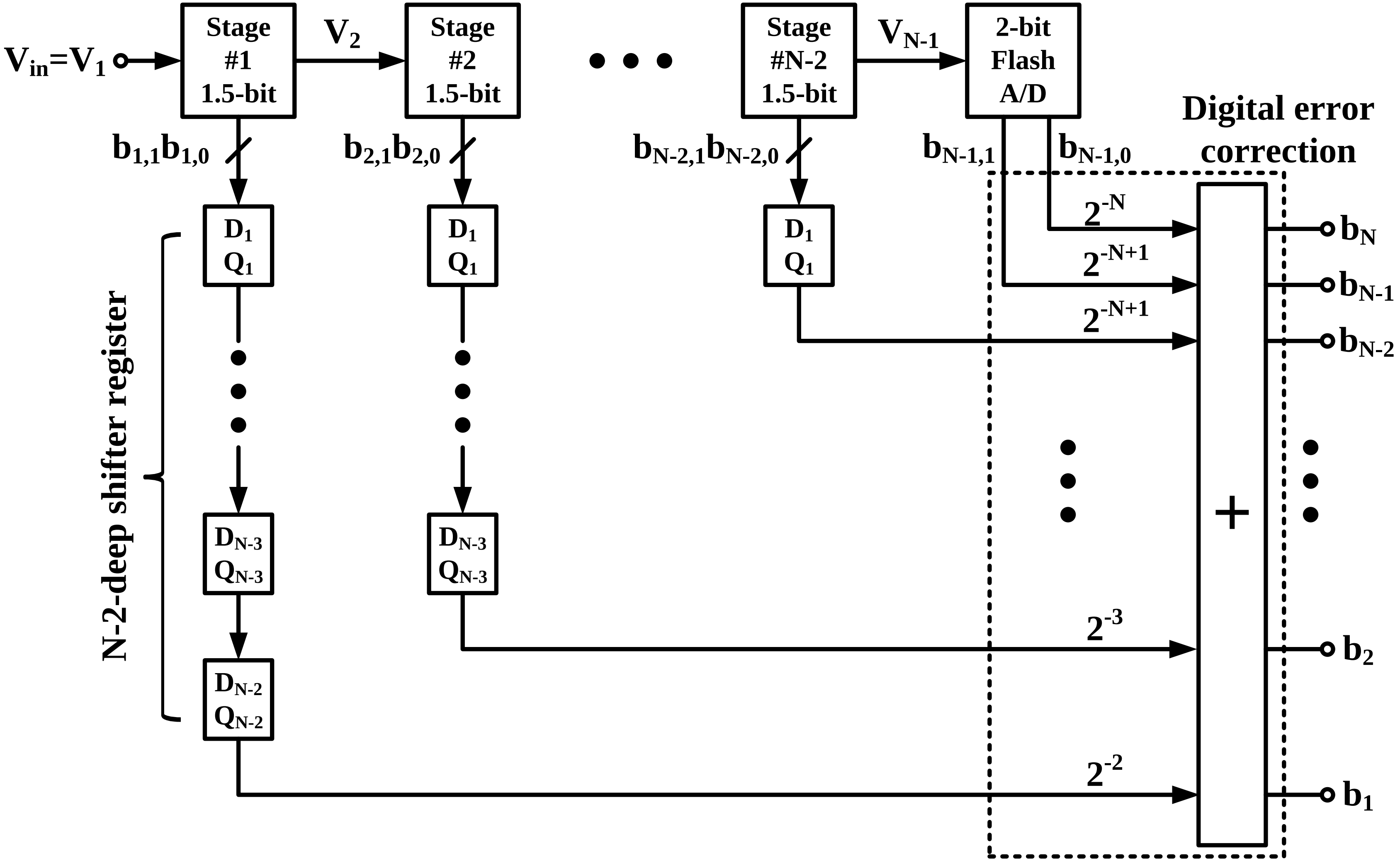
- Any comparator offset results in irreparable errors
  - ◆ 3-Bit Pipelined ADC for an example
  - ◆ Comparator of stage #1 with offset = 1/8Vref



  - ◆ For V_1=1/16V_{ref}, the digital out is $(011)_2$, but the correct digital out is $(100)_2$
- 1.5-bit/stage with digital error correction are more robust to comparator offset

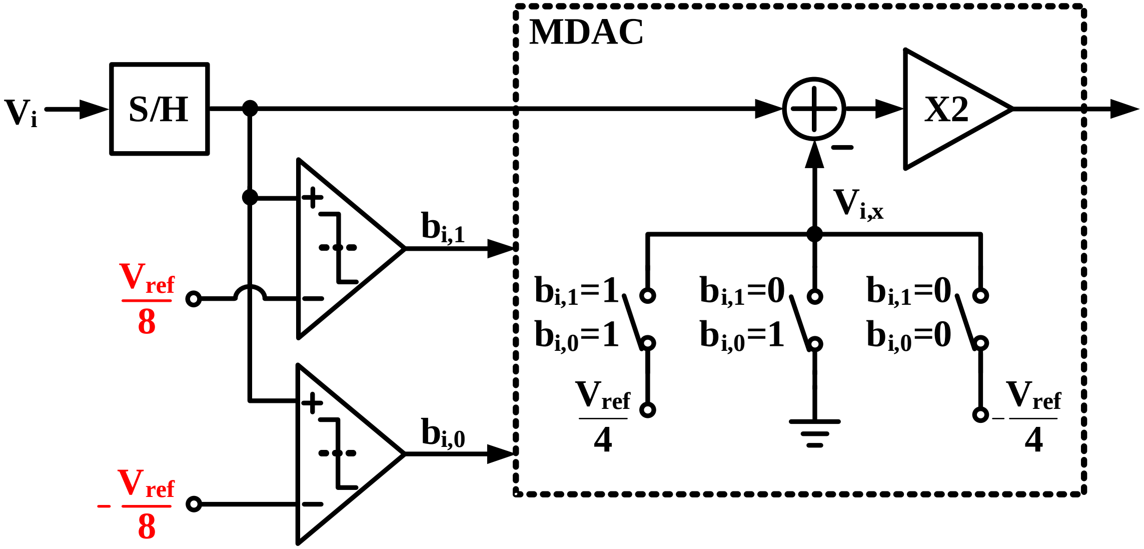
1.5-Bit/Stage Pipelined ADC

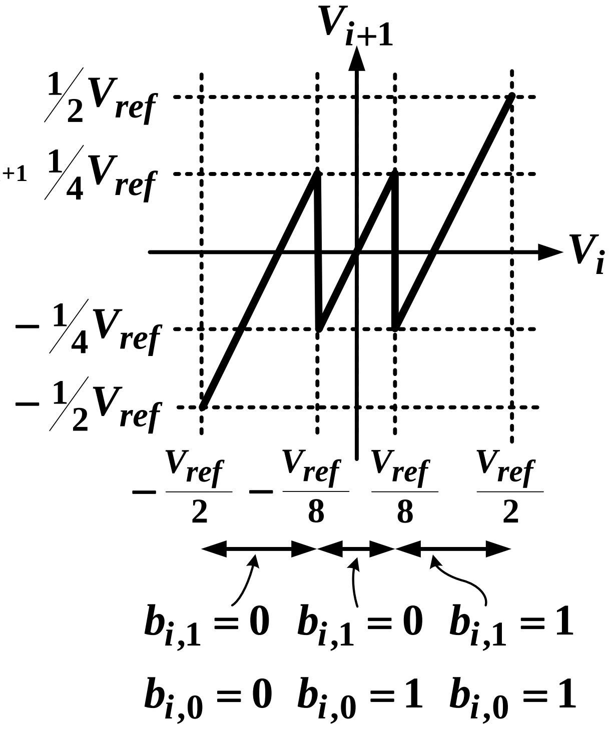
- Block diagram (with digital error correction)



1.5-Bit/Stage Pipelined ADC (Cont.)

- A 1.5-bit pipelined converter stage



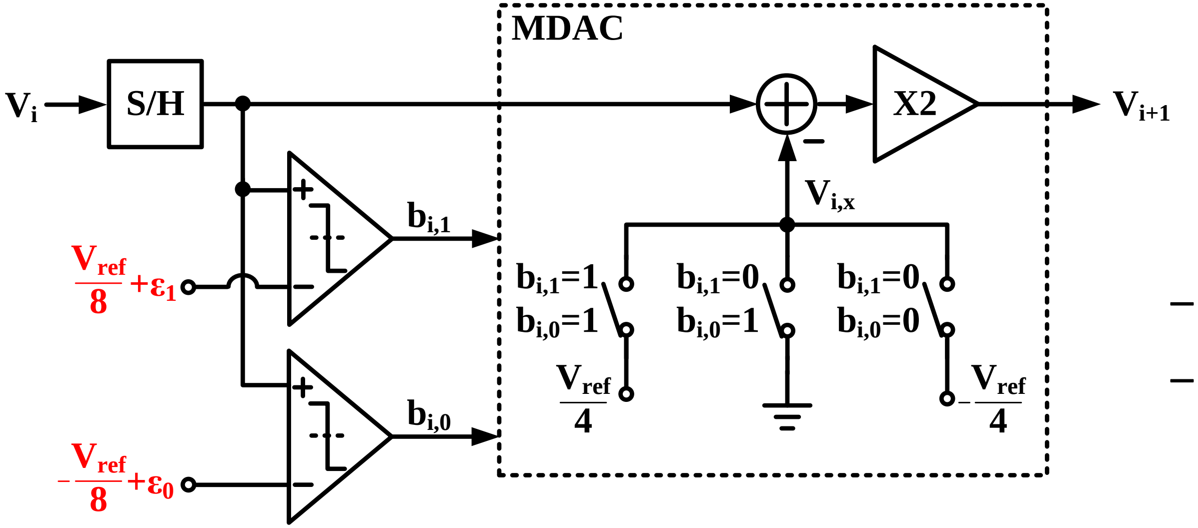


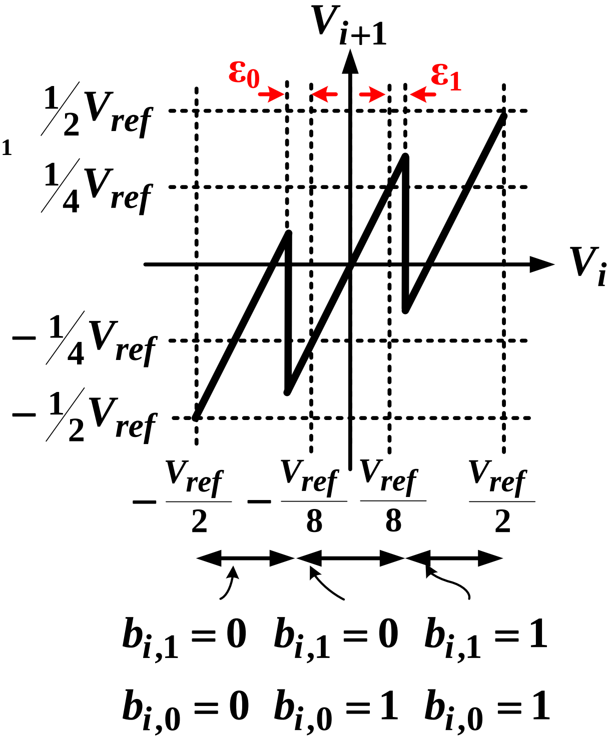
Note : $V_{i,x} = \left(b_{i,0} + b_{i,1} - 1\right)\cdot \dfrac{V_{ref}}{4}$

- ◆ Adding a second comparator to each stage → 1.5-bit/stage
- ◆ 3-level quantization of each stage input

1.5-Bit/Stage Pipelined ADC (Cont.)

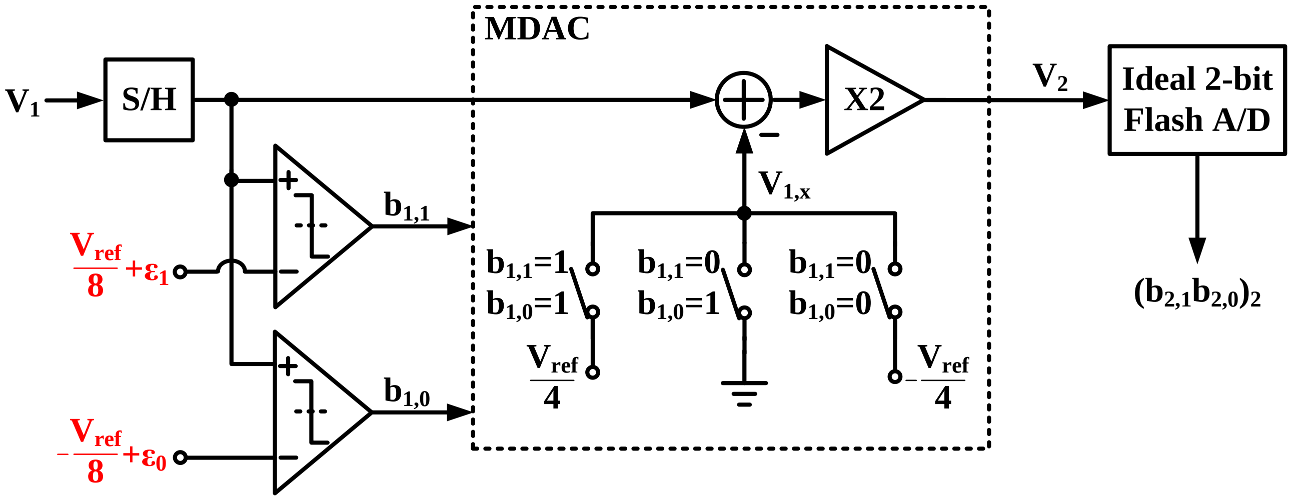
- Input-output relationship with comparator offsets





- ◆ It will introduce an error of ± V_{ref} /4 into $V_{i,x}$ (it is reflected in $b_{i,0}$ and $b_{i,1}$)
- ◆ It also result in an error of ± V_{ref} /2 in V_{i+1} (it is captured by the output bits of subsequent stages)
- ◆ When all bits are combined by the digital error correction, the errors cancel (as long as $|\varepsilon_0| < V_{ref}/8$ and $|\varepsilon_1| < V_{ref}/8$)

Example : 3-bit pipelined ADC



For $V_1 = \frac{3}{32}V_{ref}$ without offset: $\varepsilon_0 = \varepsilon_1 = 0$

$\Rightarrow$ comparator level $= \frac{1}{8}V_{ref}$ and $-\frac{1}{8}V_{ref}$

$\Rightarrow b_{1,1} = 0$ and $b_{1,0} = 1$

$\Rightarrow V_2 = 2\cdot\left(\frac{3}{32}V_{ref} - 0\right) = \frac{3}{16}V_{ref}$

$\Rightarrow b_{2,1} = 1$ and $b_{2,0} = 0$

$$\begin{array}{ll} \Rightarrow b_{1,1} & :0 \\ \quad b_{1,0} & :1 \\ +)\ b_{2,1}b_{2,0} & :10 \\ \hline \boxed{b_1b_2b_3 \quad :100} & \end{array}$$

For $V_1 = \frac{3}{32}V_{ref}$ with offset: $\varepsilon_0 = \varepsilon_1 = -\frac{1}{16}V_{ref}$

$\Rightarrow$ comparator level $= \frac{1}{16}V_{ref}$ and $-\frac{3}{16}V_{ref}$

$\Rightarrow b_{1,1} = 1$ and $b_{1,0} = 1$

$\Rightarrow V_2 = 2\cdot\left(\frac{3}{32}V_{ref} - \frac{1}{4}V_{ref}\right) = -\frac{5}{16}V_{ref}$

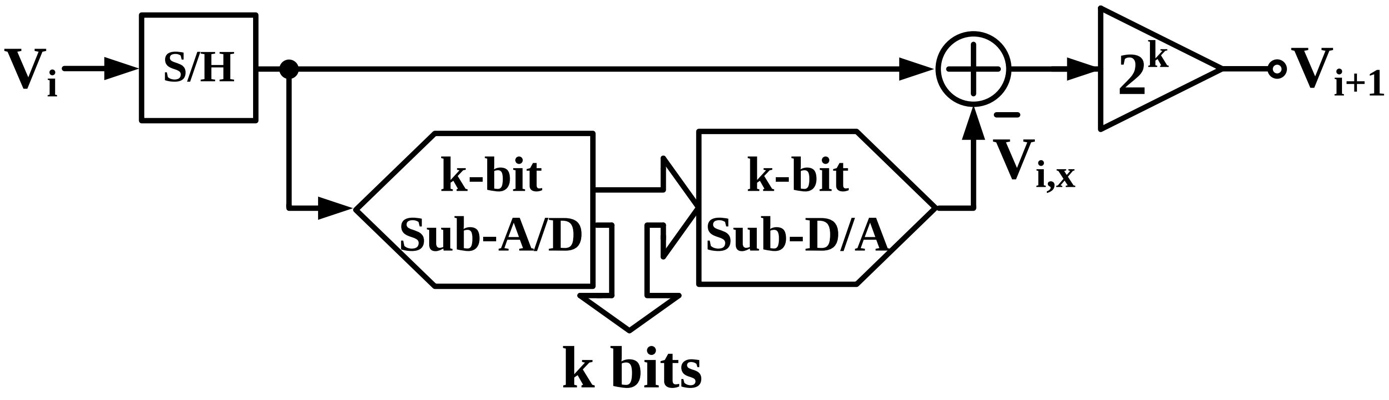
$\Rightarrow b_{2,1} = 0$ and $b_{2,0} = 0$

$$\begin{array}{ll} \Rightarrow b_{1,1} & :1 \\ \quad b_{1,0} & :1 \\ +)\ b_{2,1}b_{2,0} & :00 \\ \hline \boxed{b_1b_2b_3 \quad :100} & \end{array}$$

Digital out are the same

Multi-Bit/Stage Pipelined ADC

- Implementation
  - 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.
  - MIM capacitors are used to implement switched-capacitor of S/H gain amplifier.



Issues in Designing Pipelined ADC

- Comparator offset
  - Greatly relaxed by using digital error correction

    Reference : S. H. Lewis and P. R. Gray, “A pipelined 5-Msamples/s 9-bit analog-to-digital converter,” *IEEE J. Solid-State Circuits*, vol. 22, no. 6, pp. 954–961, Dec. 1987

- Finite opamp gain
  - Cascading gain stages/gain- boosting techniques
    - Complex circuit structures
    - Increase power consumption
  - Correlated double sampling (CDS) technique
    - For the same specification of resolution, the required gain in dB could be halved

      Reference : J. Li and U.-K. Moon, “A1.8-V 67-mW 10-bit 100-MS/s pipelined ADC using time-shifted CDS technique,” IEEE J. Solid-State Circuits, vol. 39, no. 9, pp. 1468–1476, Sep. 2004
  - Correlated level shifting (CLS) technique
    - Rail-to-rail output swing and largely relax gain requirement

      Reference : B. R. Gregoire, U. Moon, "An over-60 dB true rail-to-rail performance using correlated level shifting and an opamp with only 30 dB loop gain", IEEE J. Solid-State Circuits, vol. 43, no. 12, pp. 2620-2630, Dec. 2008.

Issues in Designing Pipelined ADC (Cont.)

- Capacitor mismatch
  - Larger capacitor sizes
    - Increase the cost and power of capacitive loads and driving circuits
  - Capacitor swapping
    - Capacitor error averaging (CEA)

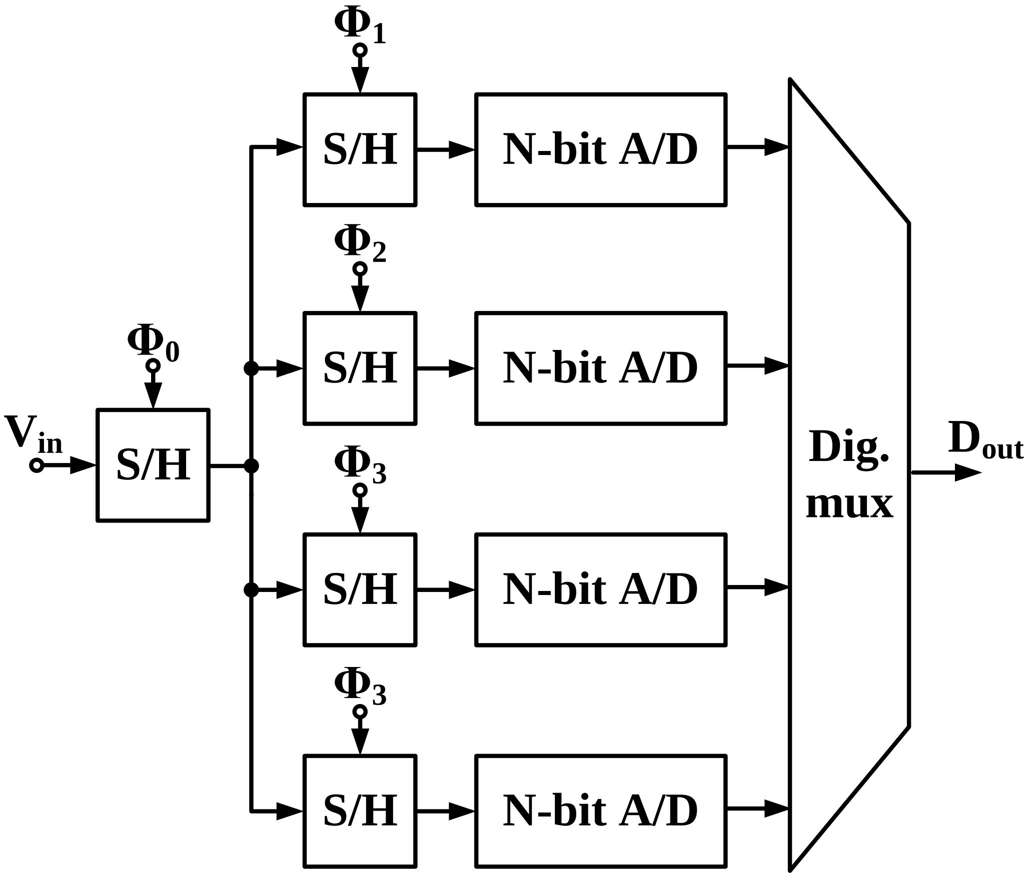
      Reference : Y. Chiu, “Inherently linear capacitor error-averaging techniques for pipelined A/D conversion,” *IEEE Trans. Circuits Syst. II*, vol. 47, no. 3, pp. 229–232, Mar. 2000
    - Commutated Feedback Capacitor Switching (CFCS)

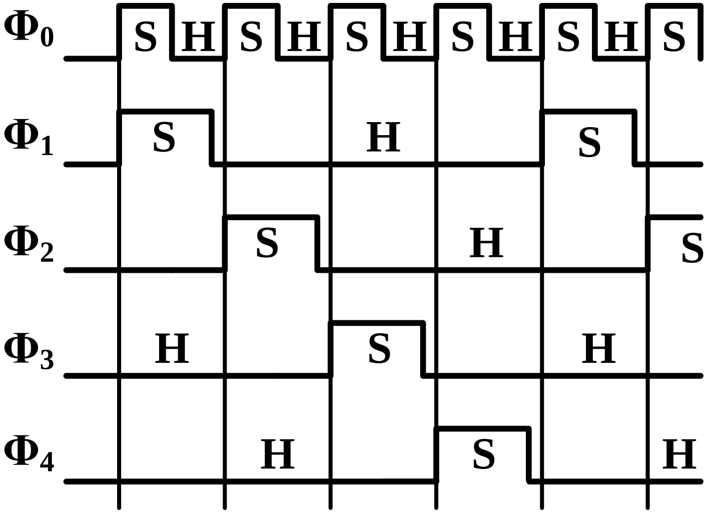
      Reference : P. C. Yu and H.-S. Lee, “A 2.5-V,12-b, 5-Msample/s pipelined CMOS ADC,” *IEEE J. Solid-State Circuits*, vol. 31, no. 12, pp. 1854–1861, Dec. 1996
    - Random feedback-capacitor interchanging (RFCI)
    - Averaging RFCI (ARFCI)

      Reference : C.-H. Kuo and T.-H Kuo, “Capacitor-swapping cyclic A/D conversion techniques with reduced mismatch sensitivity,” *IEEE Trans. Circuits Syst. II*, vol. 55, no. 12, pp. 1219–1223, Dec. 2008
  - Digital calibration

Time-Interleaved ADC

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





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.

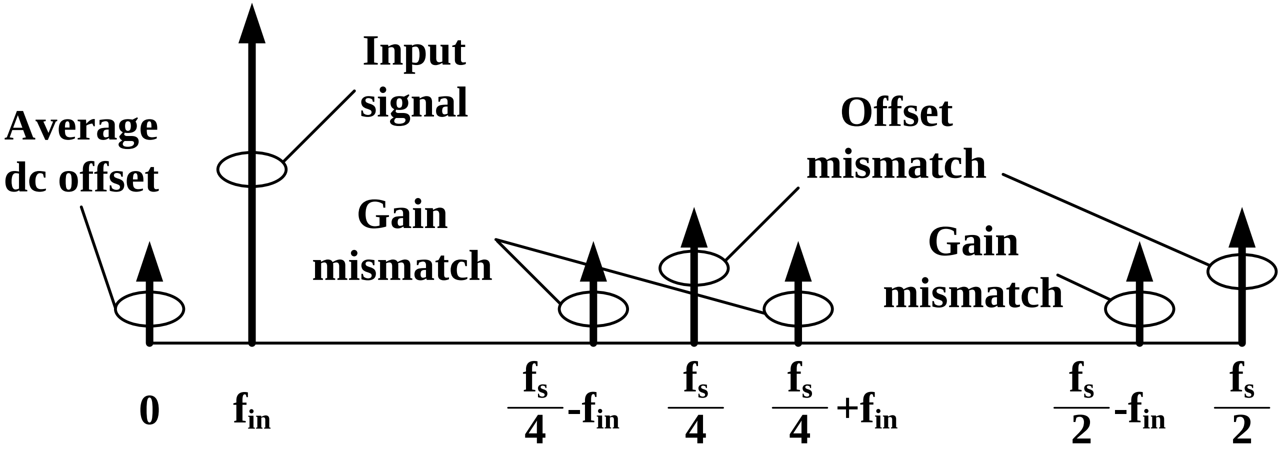
  Sometimes, the input S/H is realized in different technology, such as GaAs, while the remaining S/H circuits could be realized in silicon.
- It is also essential that the channels are extremely well matched, as mismatches will produce tones.

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

Time-Interleaved ADC (Cont.)

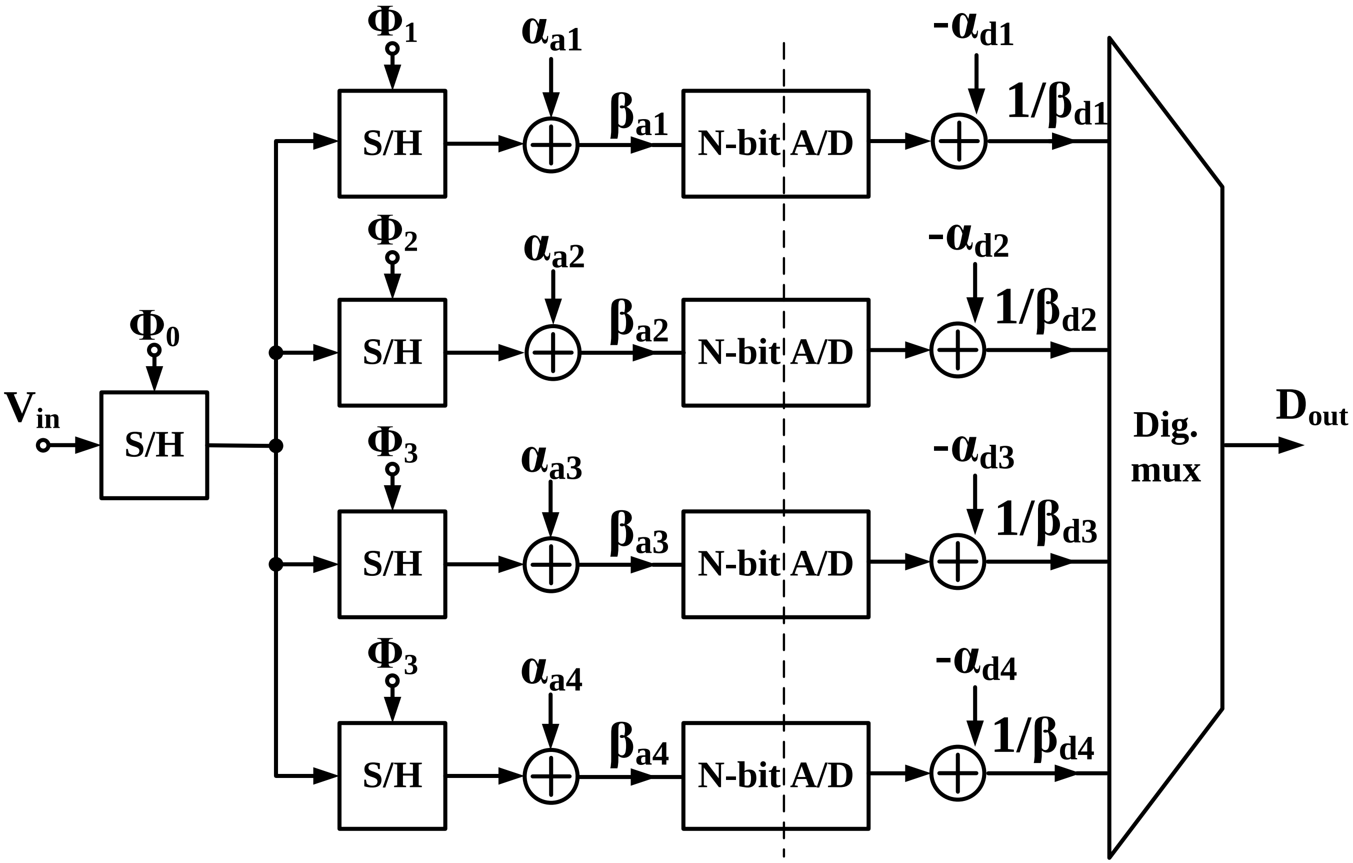
- Mismatch effect for an m-channels time-interleaved ADC
  - Offset mismatch → tones at f_s/m
    - They are independent of input frequency or amplitude (they will be present even if the input is zero)
  - Gain mismatch → tones at $kf_s/m \pm f_{in}$ for integers k
    - Their frequency and amplitude depends upon input frequency and amplitude

- Output spectrum of a 4-channels time-interleaved ADC



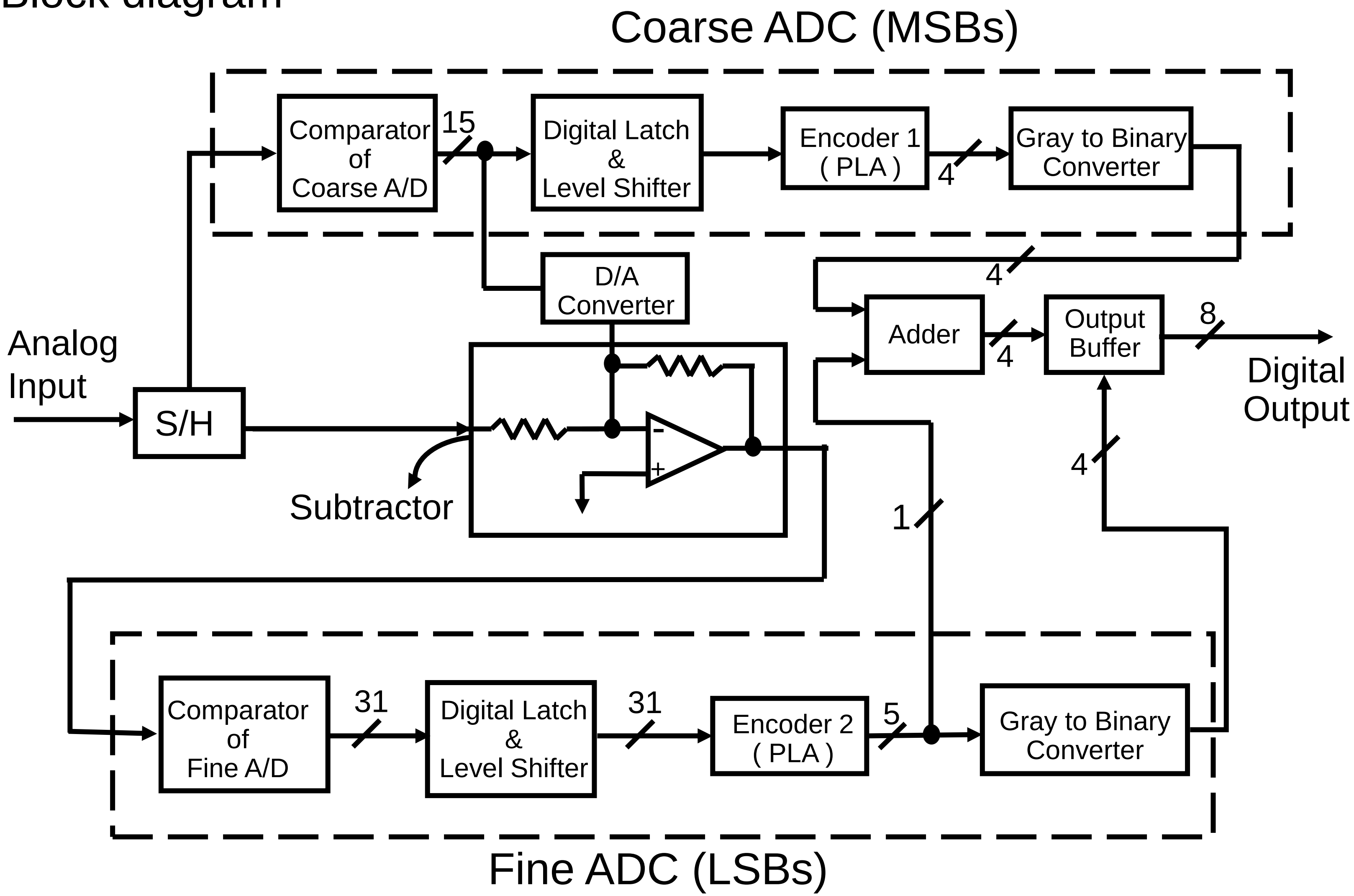
Time-Interleaved ADC (Cont.)

- Calibration of offset mismatch α_{ai} and gain mismatch β_{ai}
  - Initially, α_{di}=0 and β_{di}=0
  - V_{in}=0 → obtain $\alpha_{di} \approx \alpha_{ai}$
  - V_{in}= V_{ref} → obtain $\beta_{di} \approx \beta_{ai}$



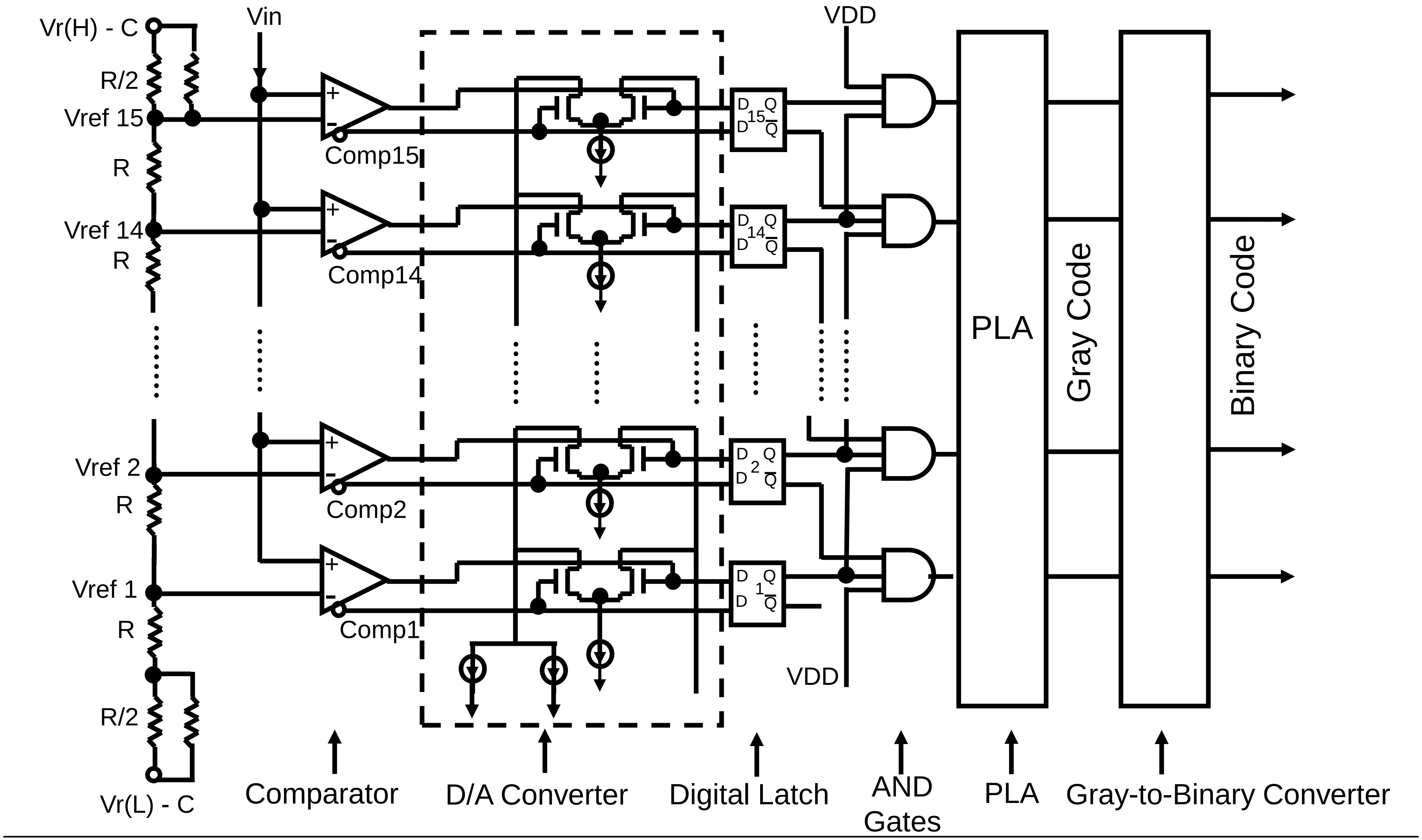
Appendix : Implementation of an 8-bit Two-Step (or Subrange) ADC

- Block diagram



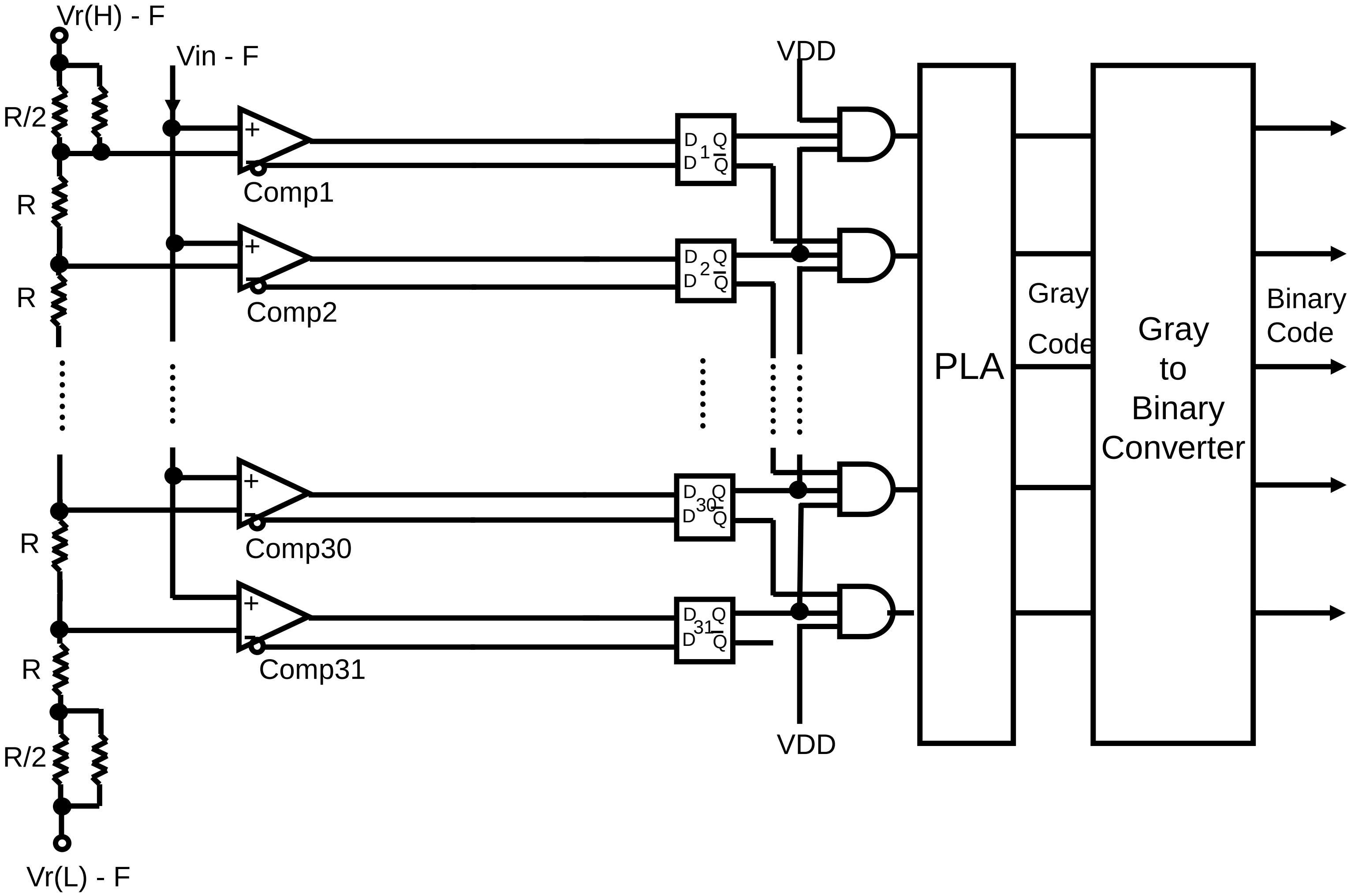
Appendix : Implementation of an 8-bit Two-Step (or Subrange) ADC (Cont.)

- Coarse ADC structure



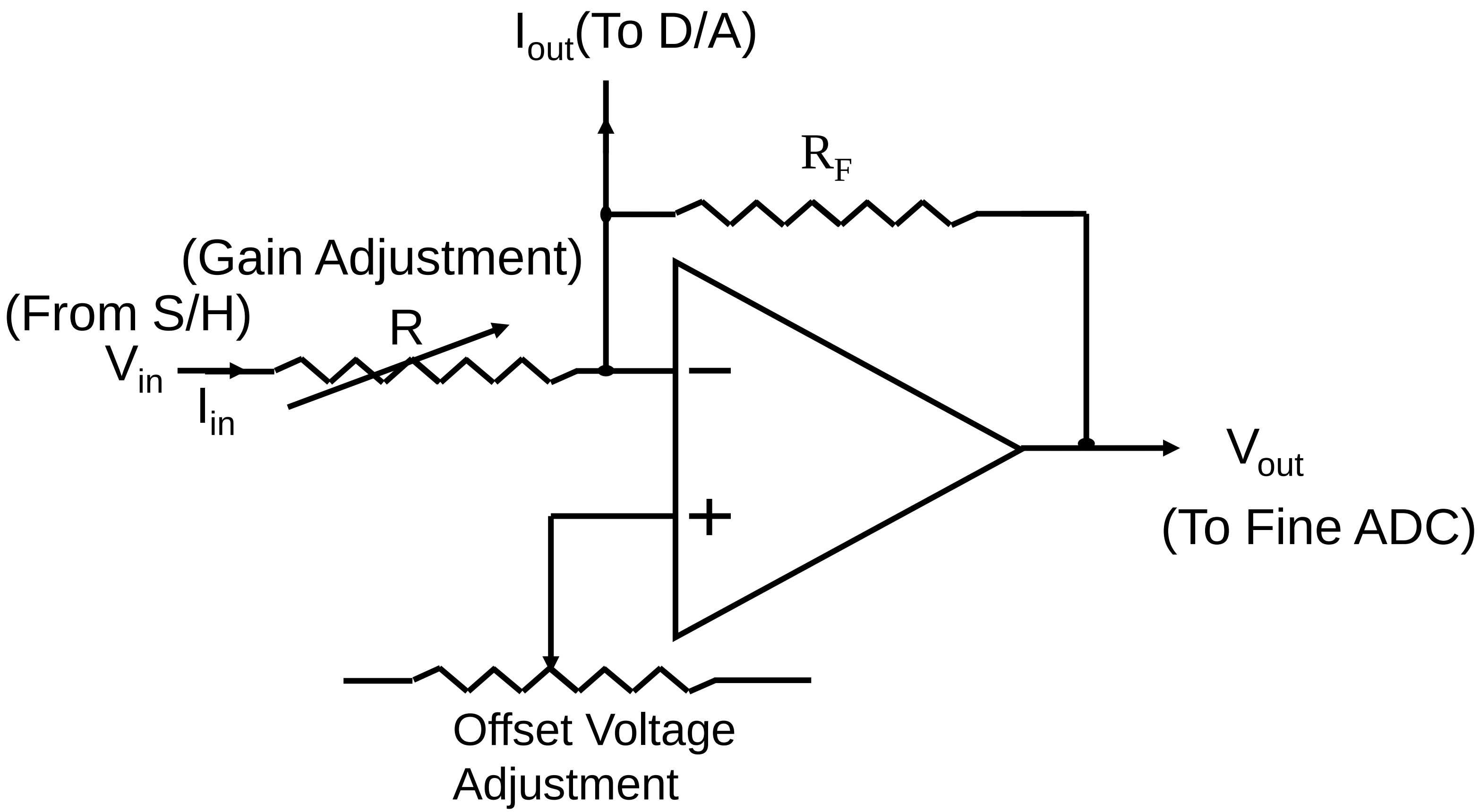
Appendix : Implementation of an 8-bit Two-Step (or Subrange) ADC (Cont.)

- Fine A/D structure



Appendix : Implementation of an 8-bit Two-Step (or Subrange) ADC (Cont.)

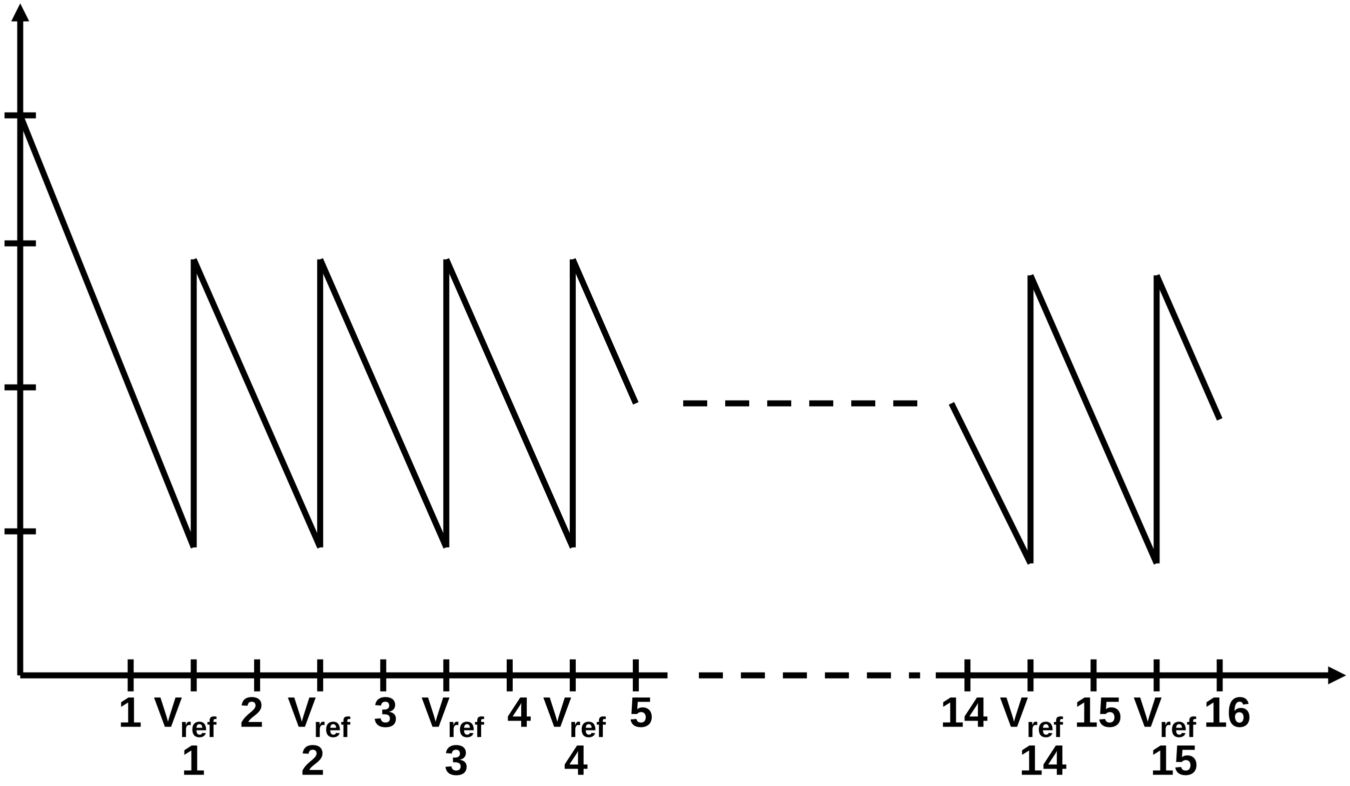
- Subtractor



Appendix : Implementation of an 8-bit Two-Step (or Subrange) ADC (Cont.)

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

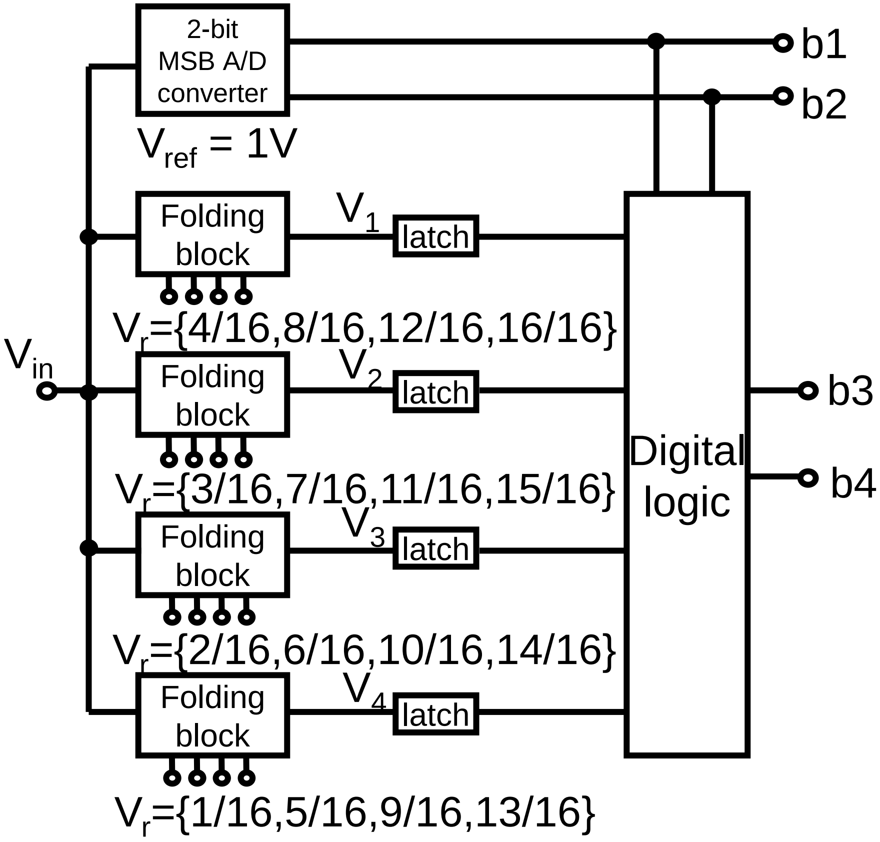
Input Voltage To Fine ADC
(i.e., Subtractor Output)



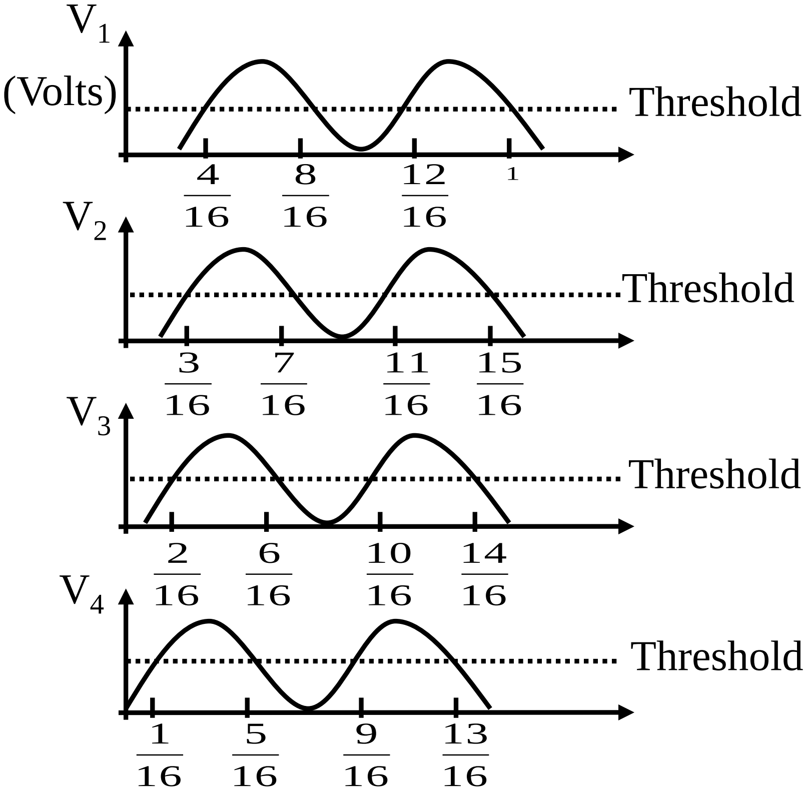
Input Voltage To Coarse ADC (i.e.S/H Output)

Appendix: Folding ADC

- 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.

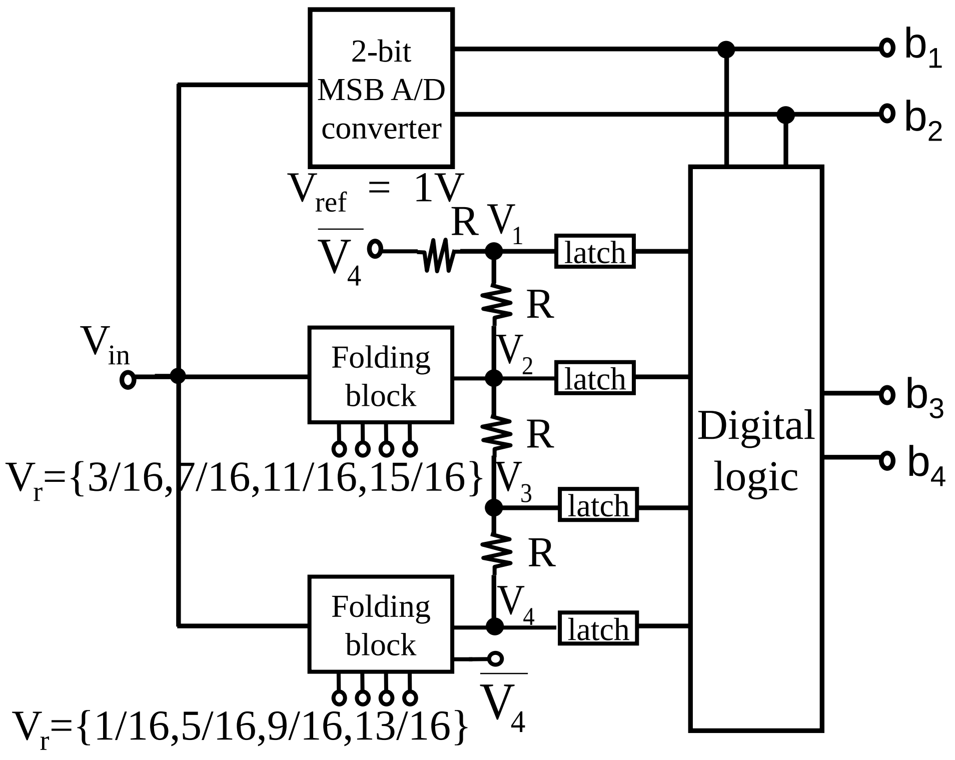


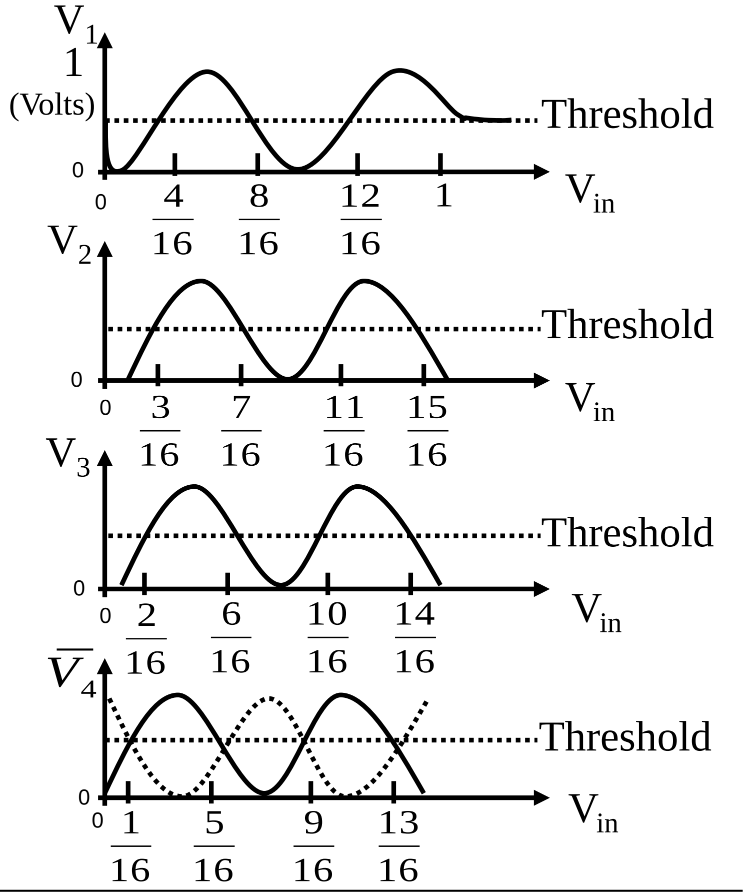
Folding block responses



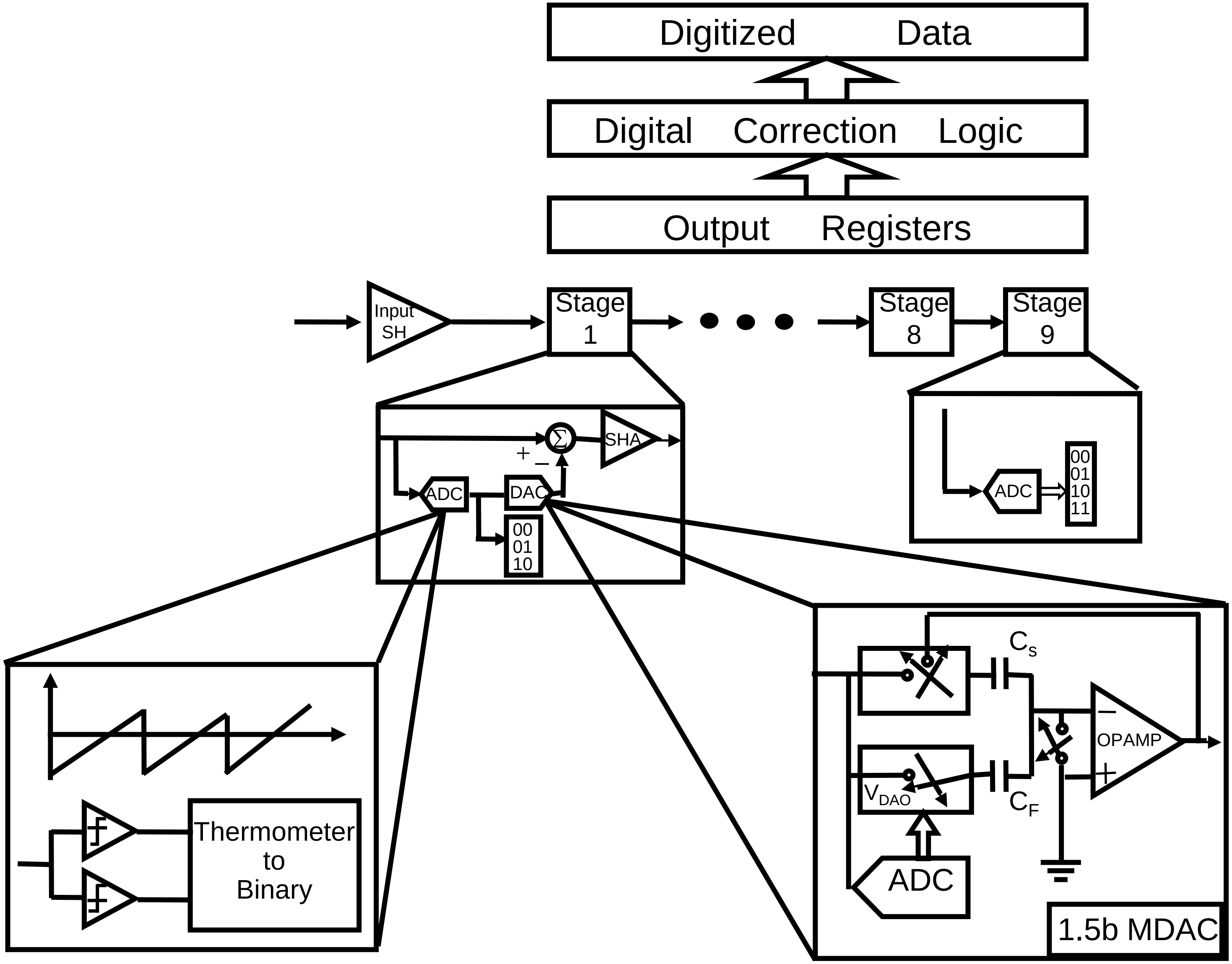
Appendix: Folding ADC

- 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





Appendix : 1.5b/Stage Pipeline Architecture


Digitized Data
Digital Correction Logic
Output Registers
Input SH
Stage 1
Stage 8
Stage 9
Σ
SHA
+
–
ADC
DAC
00
01
10
ADC
00
01
10
11
Thermometer to Binary
C_s
C_F
V_{DAO}
OPAMP
–
+
ADC
1.5b MDAC