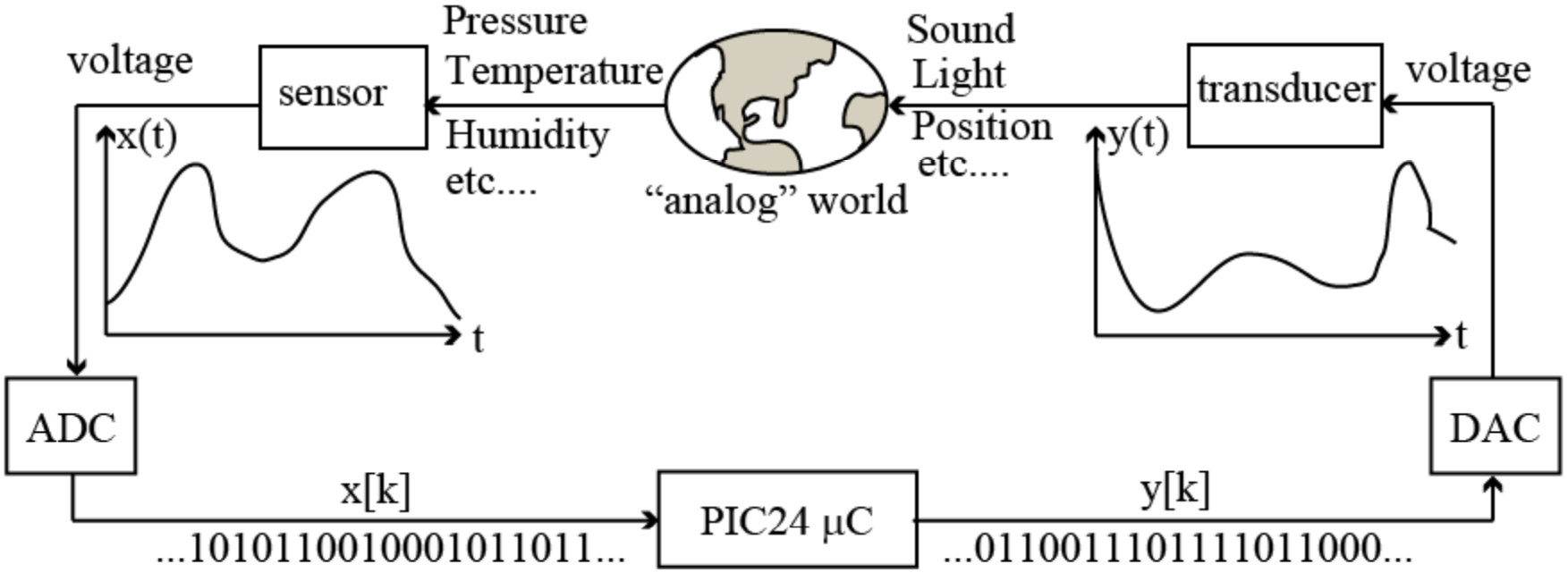
Digital Signal Processing



Analog-to-Digital Converter (ADC) converts an input analog value to an output digital representation.

This digital data is processed by a microprocessor and output to a Digital-to-Analog Converter (DAC) the converts an input binary value to an output voltage.

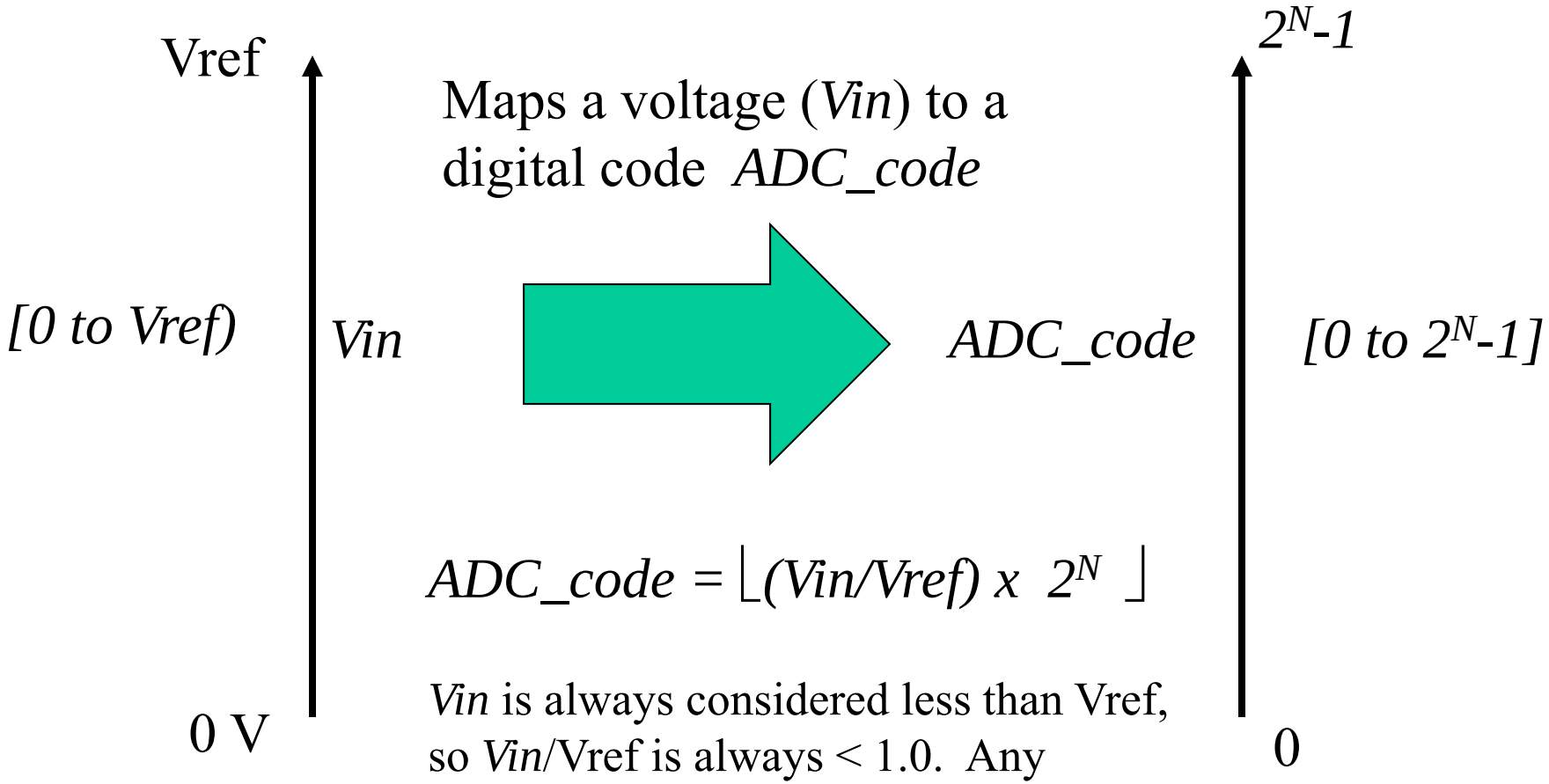
Applications

- Audio
  - Speech recognition
  - special effects (reverb, noise cancellation, etc)
- Video
  - Filtering
  - Special effects
  - Compression
- Data logging

Vocabulary

- ADC (Analog-to-Digital Converter) – converts an analog signal (voltage/current) to a digital code
- DAC (Digital-to-Analog Converter) – converts a digital code to an analog value (voltage/current)
- Sample period – for ADC, time between each conversion
  - Typically, samples are taken at a fixed rate
- Vref (Reference Voltage) – analog signal varies between 0 and Vref, or between +/- Vref
- Resolution – number of bits used for conversion (8 bits, 10 bits, 12 bits, 16 bits, etc).
- Conversion Time – the time it takes for an analog-to-digital conversion

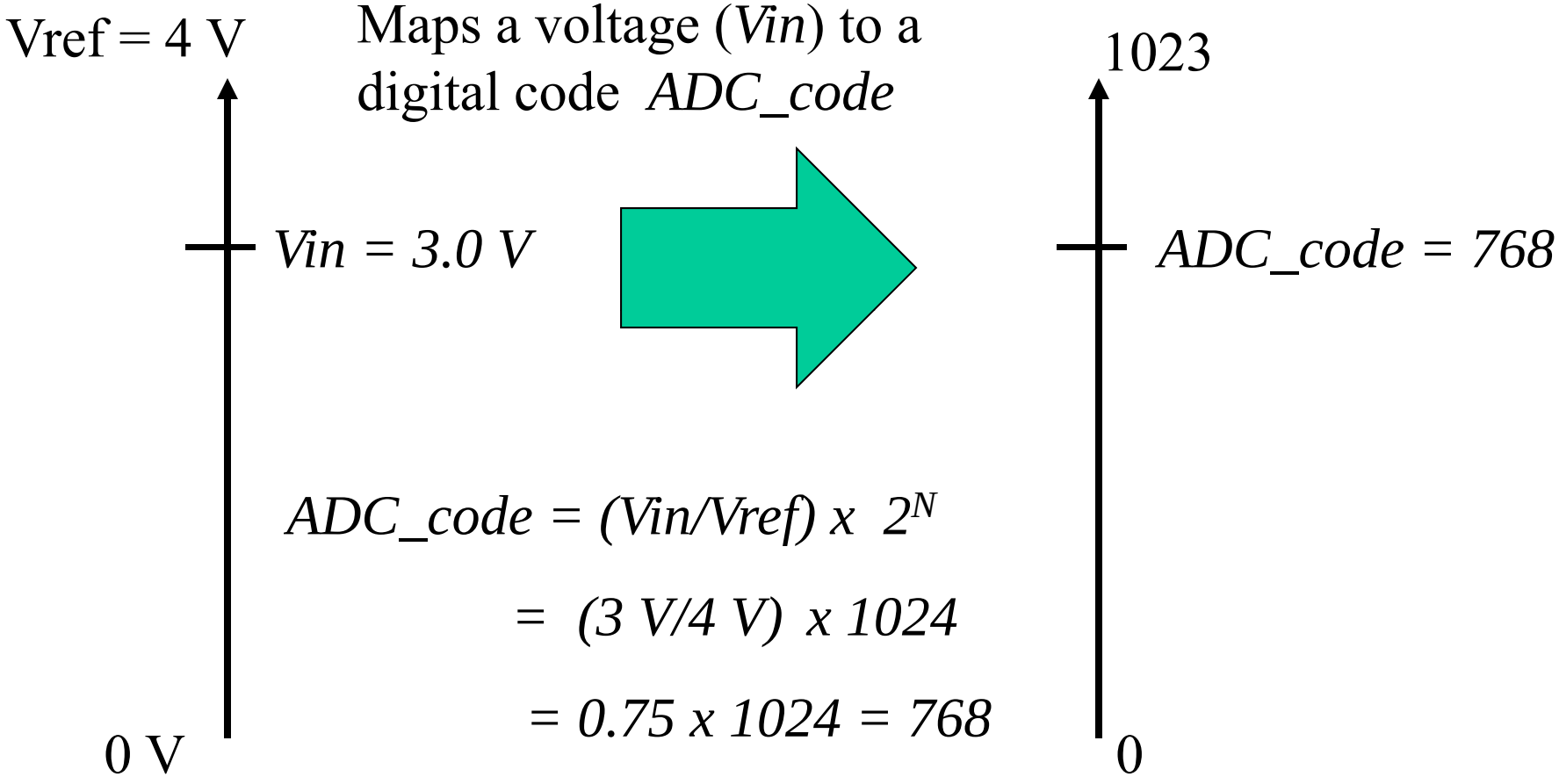
An N-bit ADC



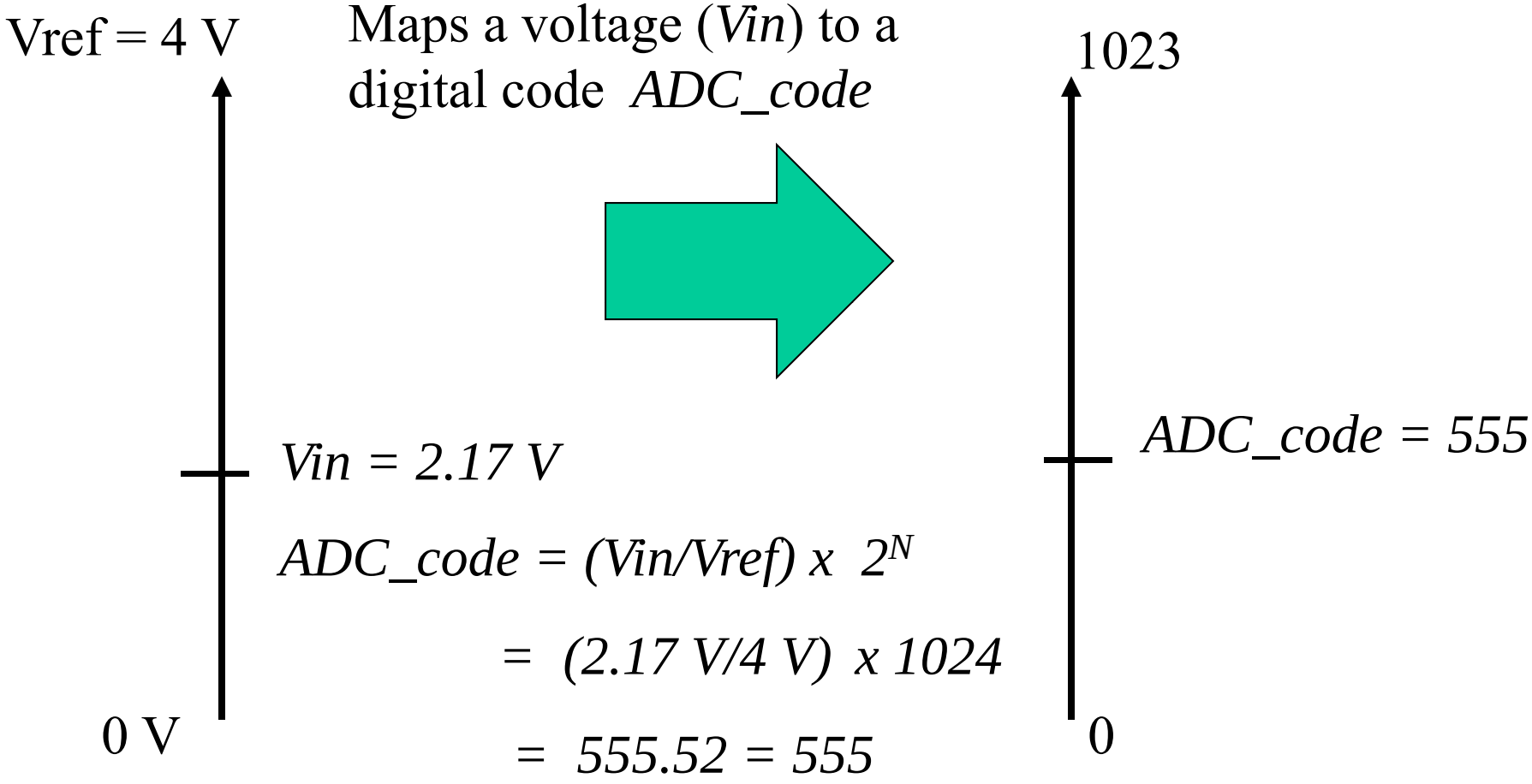
$$ADC\_code = \lfloor (Vin/Vref) \times 2^N \rfloor$$

Vin is always considered less than Vref, so *Vin*/Vref is always < 1.0. Any fractional part of the code is truncated.

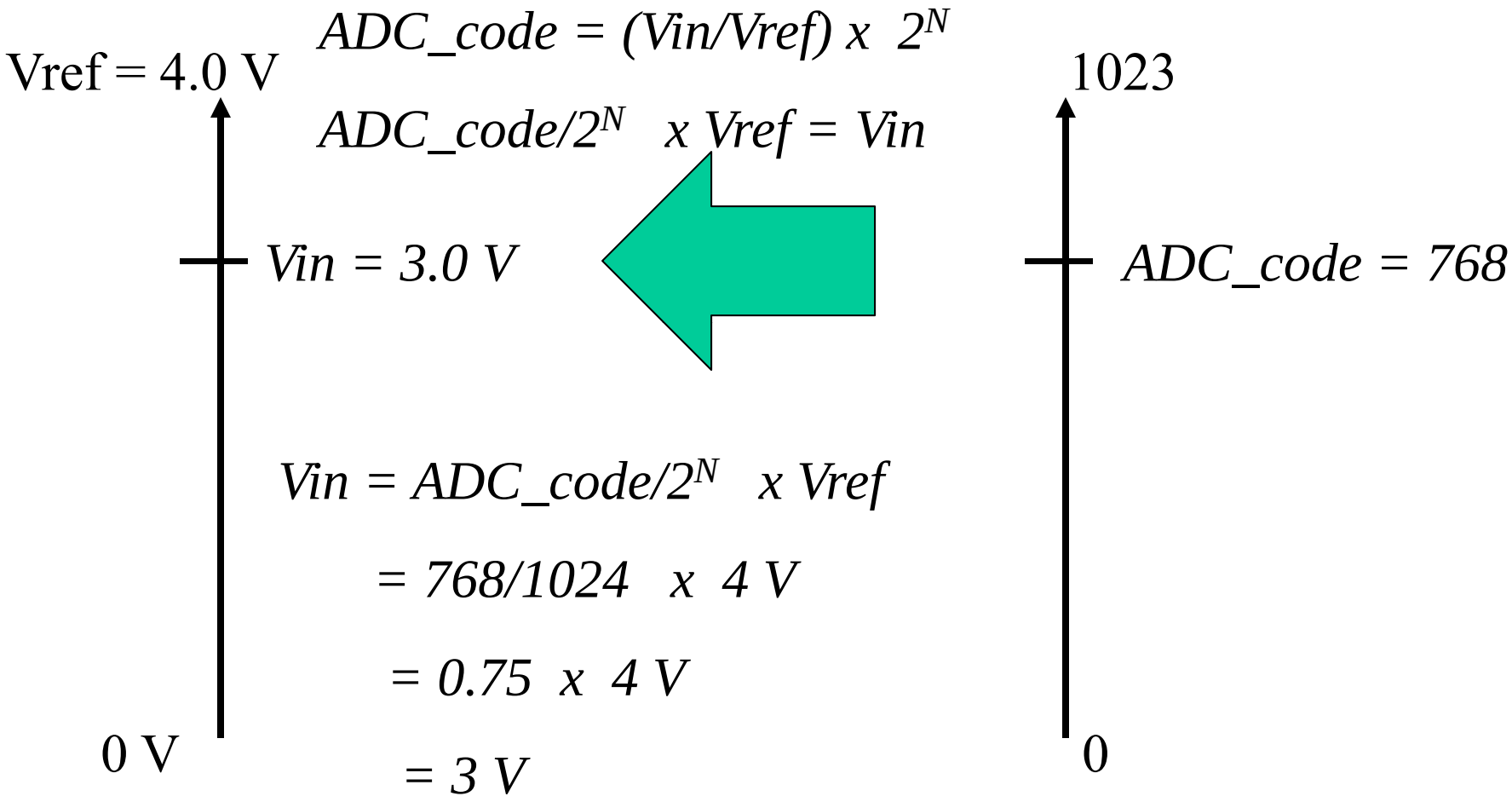
Example: A 10-bit ADC



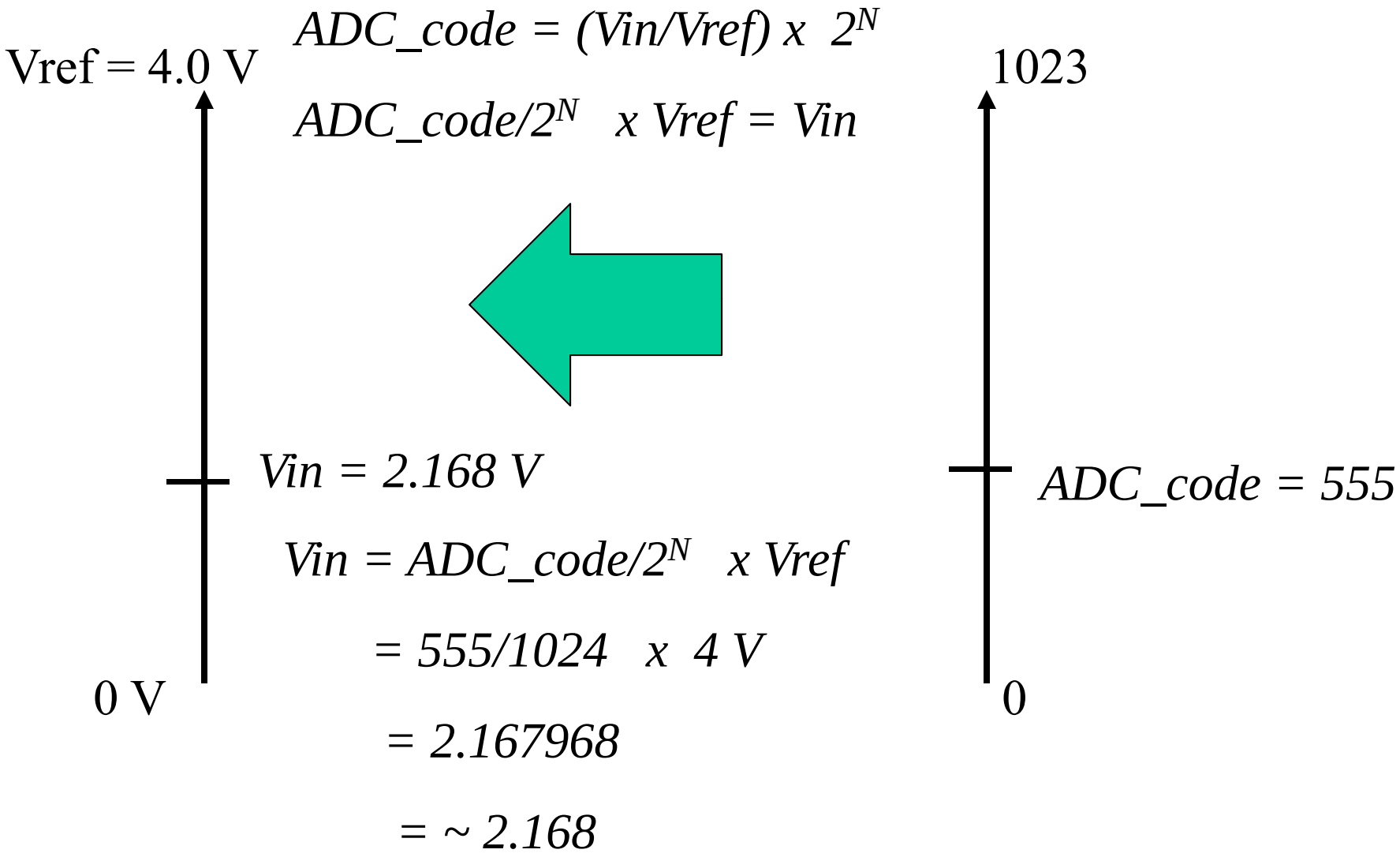
Example: A 10-bit ADC



Going from Code to Voltage



Going from Code to Voltage



ADC Resolution

For an N-bit ADC, the smallest input voltage that can be resolved is 1 LSb, or:

$1/2^N * (Vref+ - Vref-)$

Where Vref+ is the positive reference voltage and
Vref- is the negative reference voltage.

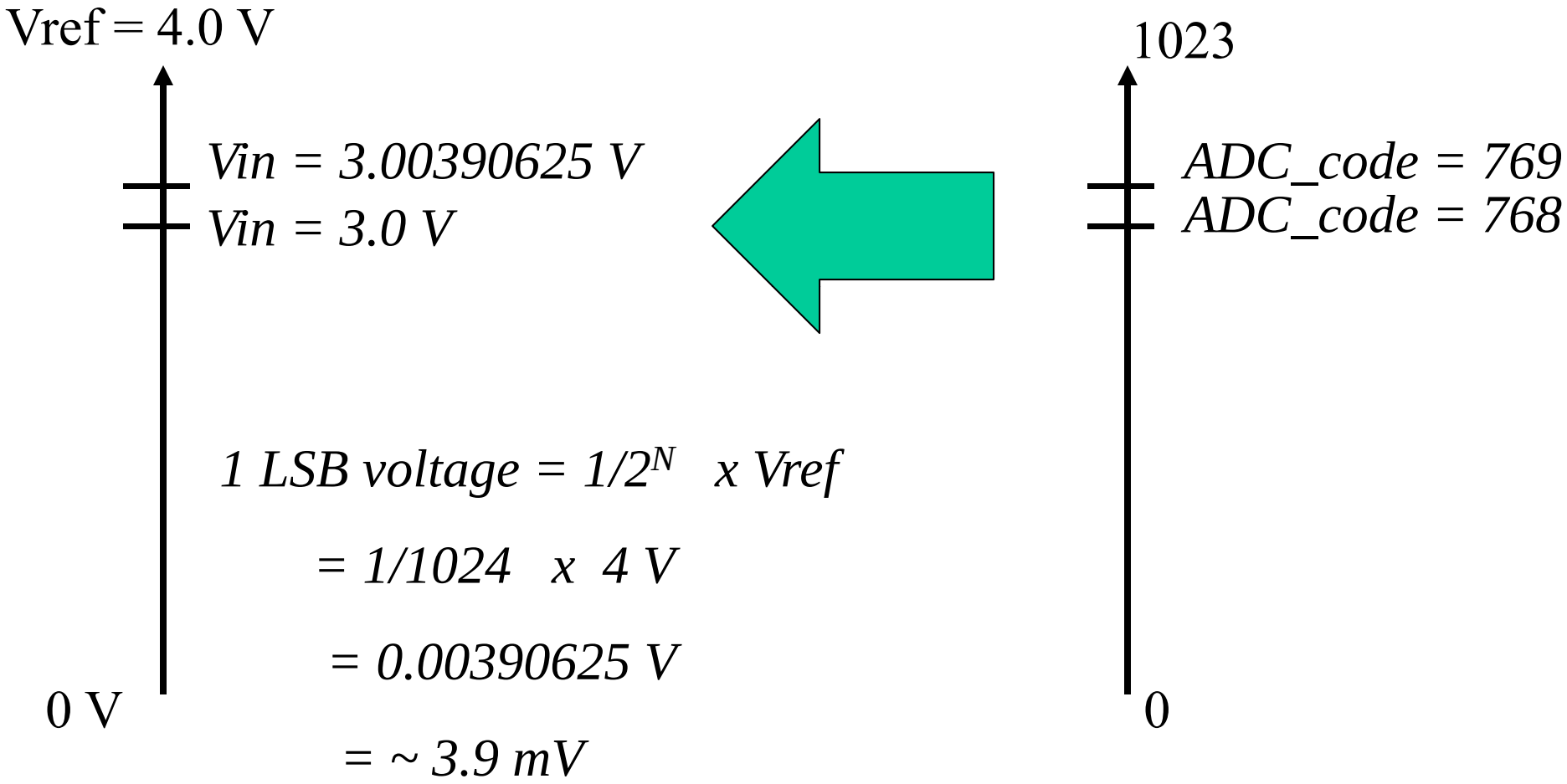
We will use Vref- = 0 V, and refer to Vref+ as simply Vref, so this simplifies to

$1/2^N * Vref.$

For Vref = 4 V, and N = 4, what is 1 LSb?

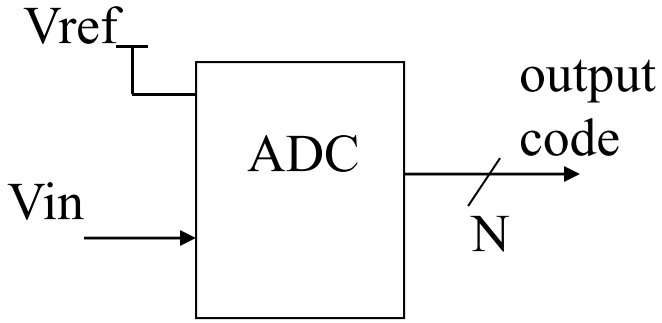
$1/2^4 * 4\ V = 1/16 * 4\ V = 0.25\ V.$

Example: 10-bit ADC Resolution



ADC, DAC Equations

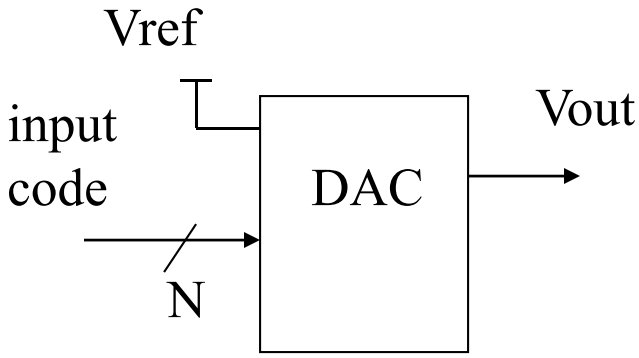
ADC: Vin = input voltage, Vref+ = reference voltage, Vref- = 0 V.
N = number of bits of precision



$$Vin/ Vref * 2^N = output\_code$$
$$output\_code/ 2^N * Vref = Vin$$

$$1\ LSB = Vref/2^N$$

DAC: Vout = output voltage, Vref = reference voltage,
N = number of bits of precision



$$Vout/ Vref * 2^N = input\_code$$
$$input\_code/ 2^N * Vref = Vout$$

$$1\ LSB = Vref/2^N$$

Sample ADC, DAC Computations

If Vref = 5V, and a 10-bit A/D output code is 0x12A, what is the ADC input voltage?

Vin = output_code/2^N * Vref = (0x12A)/2^{10} * 5 V

= 298/1024 * 5 V = 1.46 V (ADC Vin)

If Vref = 5V, and an 8-bit DAC input code is 0xA9, what is the DAC output voltage?

Vout = input_code/2^N * Vref = (0xA9)/2^8 * 5 V

= 169/256 * 5 V = 3.3 V (DAC Vout)

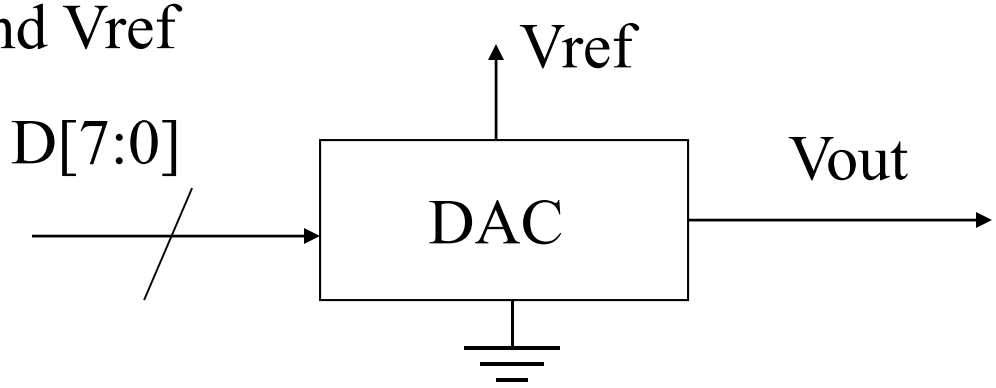
If Vref = 4V, and an 8-bit A/D input voltage is 2.35 V, what is the ADC output code?

output code = Vin/ Vref * 2^N = 2.35 V/ 4 V * 2^8

= .5875 * 256 = 150.4 = 150 = 0x96 (ADC output code)

Digital-to-Analog Conversion

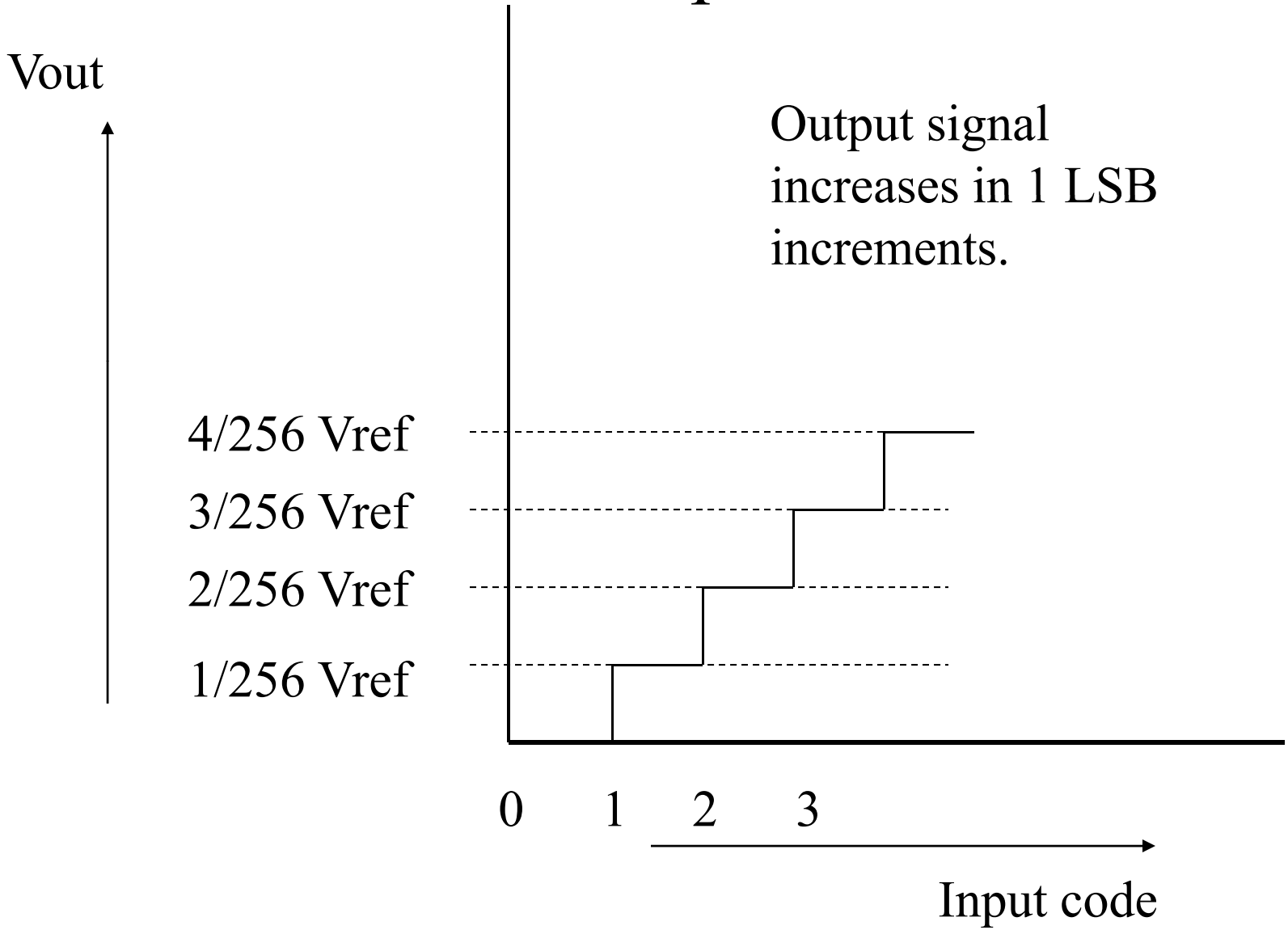
For a particular binary code, output a voltage between 0 and Vref



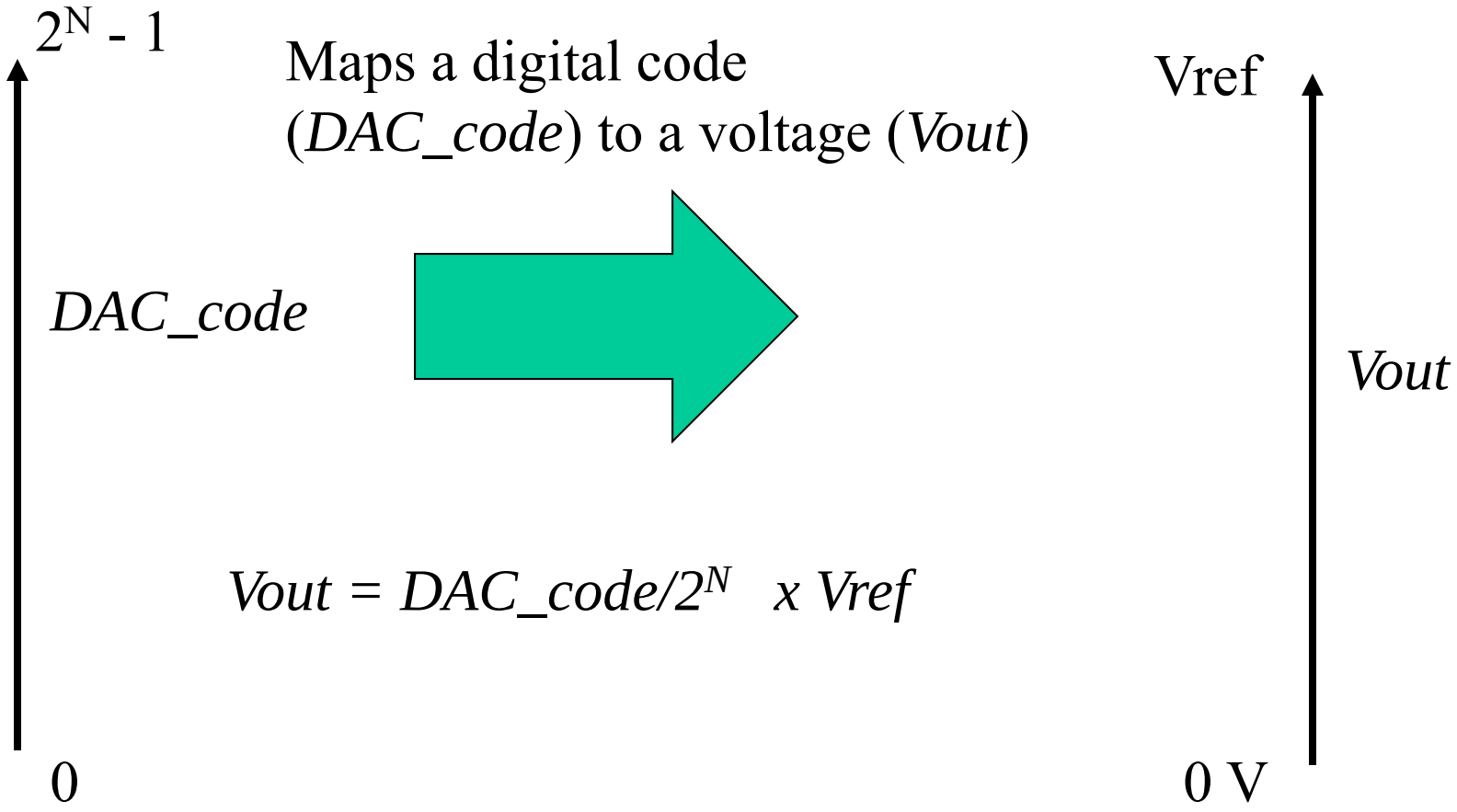
Assume a DAC that uses an unsigned binary input code, with $0 \leq$ Vout < Vref. Then

D= 0000 0000 Vout = 0V
D= 0000 0001 Vout = Vref(1/256) (one LSB)
D = 0000 0010 Vout = Vref(2/256)
...
D = 1111 1111 Vout = Vref(255/256) (full scale)

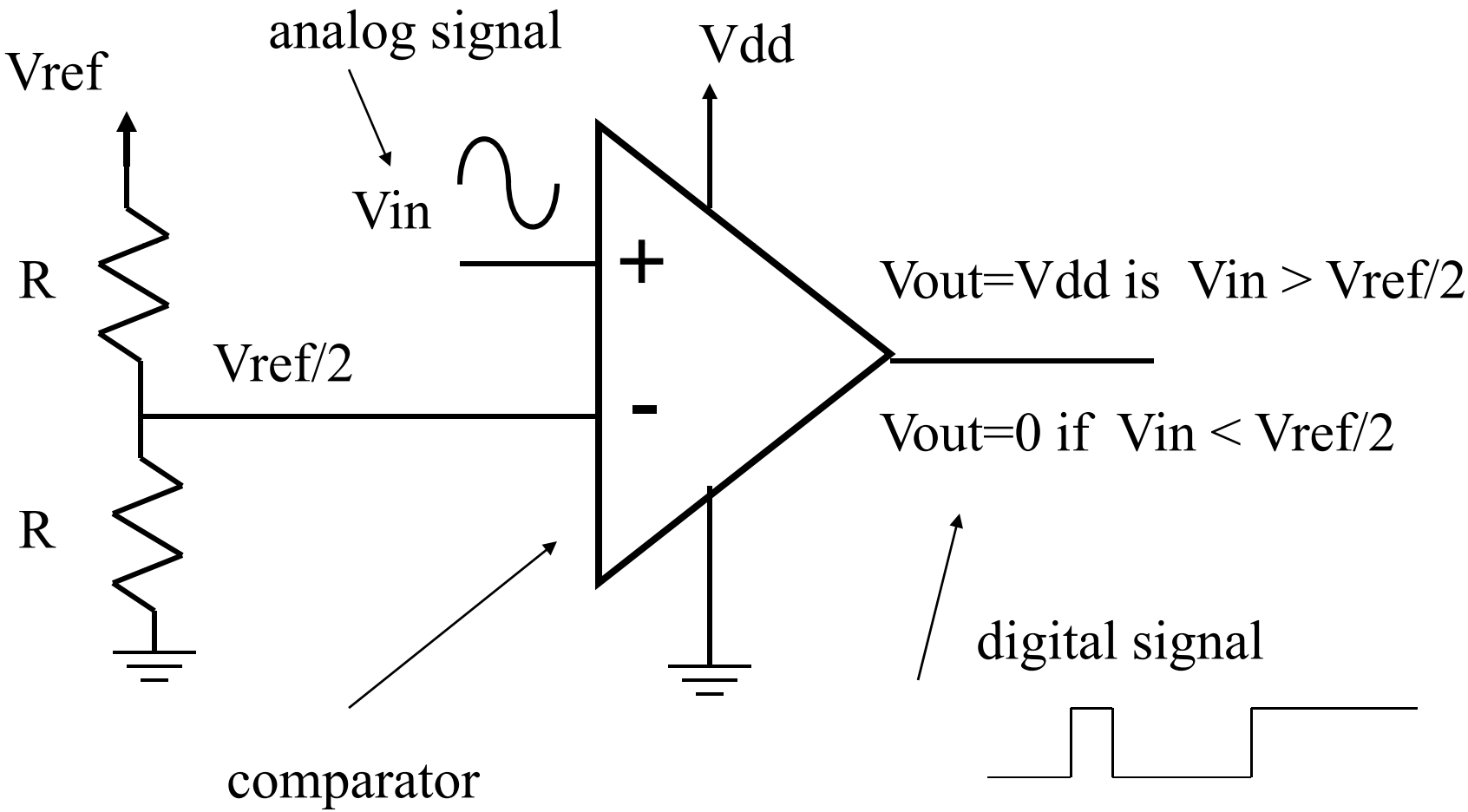
DAC Output Plot



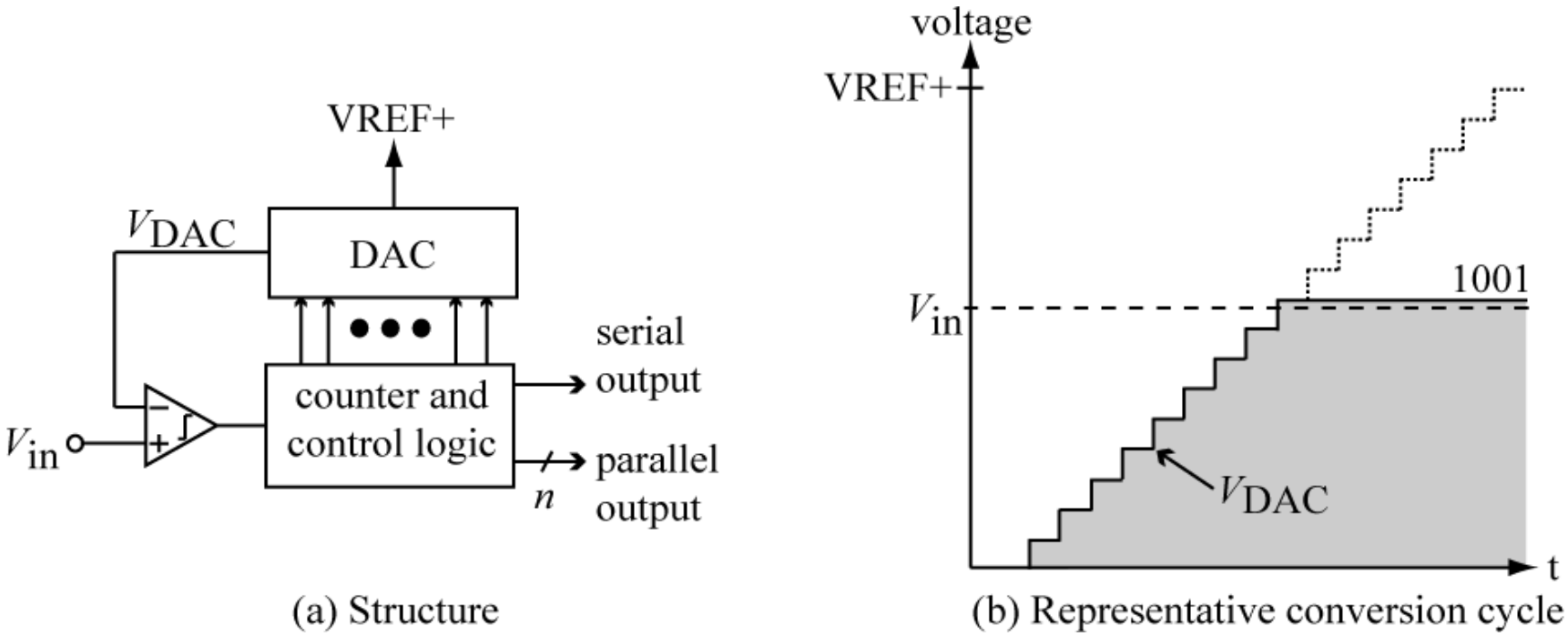
An N-bit DAC


$2^N - 1$
Maps a digital code
(*DAC_code*) to a voltage (*Vout*)
Vref
DAC_code
Vout
$Vout = DAC\_code/2^N \quad x\ Vref$
0
0 V

A 1-bit ADC

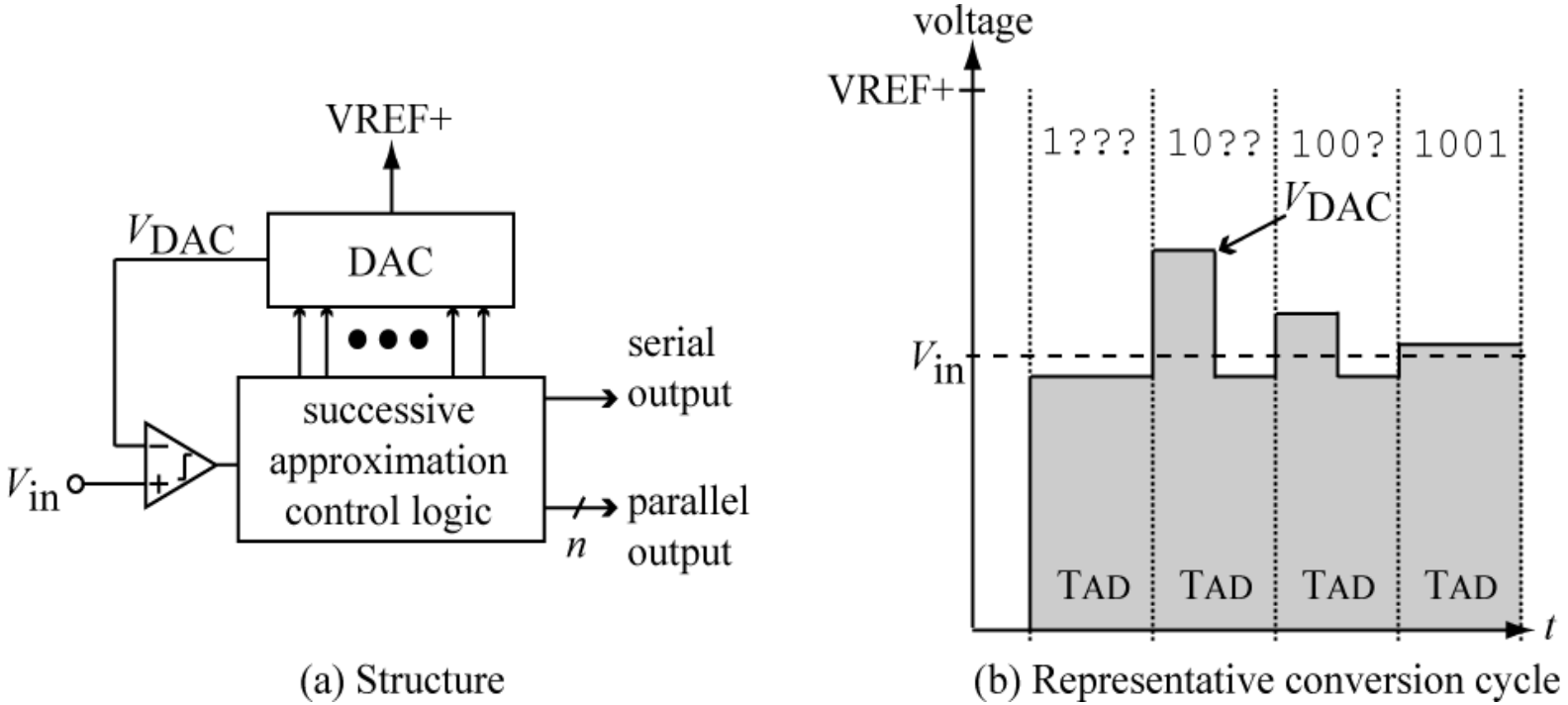

analog signal
Vdd
Vref
Vin
R
+
Vout=Vdd is Vin > Vref/2
Vref/2
-
Vout=0 if Vin < Vref/2
R
digital signal
comparator

Counter Ramp ADC



Control logic use a counter to apply successive codes 0,1,2,3,4... to DAC (Digital-to-Analog Converter) until DAC output is greater than Vin. This is SLOW, and have to allocate the worst case time for each conversion, which is 2^N clock cycles for an N-bit ADC.

Successive Approximation ADC



Initially set VDAC to ½ Vref, then see if Vin higher or lower tan VDAC. If > ½ Vref, then next guess is between Vref and ½ Vref, else next guess is between ½ Vref and GND. Do this for each bit of the ADC. Takes N clock cycles.

Successive Approximation Example

Given a 4-bit Successive Approximation ADC, and Vref = 4 V.
Let Vin = 3.14159 V. Clear DAC input to 0b0000.

1. First guess, DAC input = 0b1000 = 8, so Vdac = $8/2^4$* 4 V = 8/16 * 4 V = 2 V.
   Vdac (2 V) < Vin (3.14159 V), so guess of '1' for MSb of DAC was correct.

2. Set next bit of DAC to '1', DAC input = 0b1100 = 12, so Vdac = 12/16*4= 3V.
   Vdac (3 V) < Vin (3.14159 V), so guess of '1' for bit2 of DAC was correct.

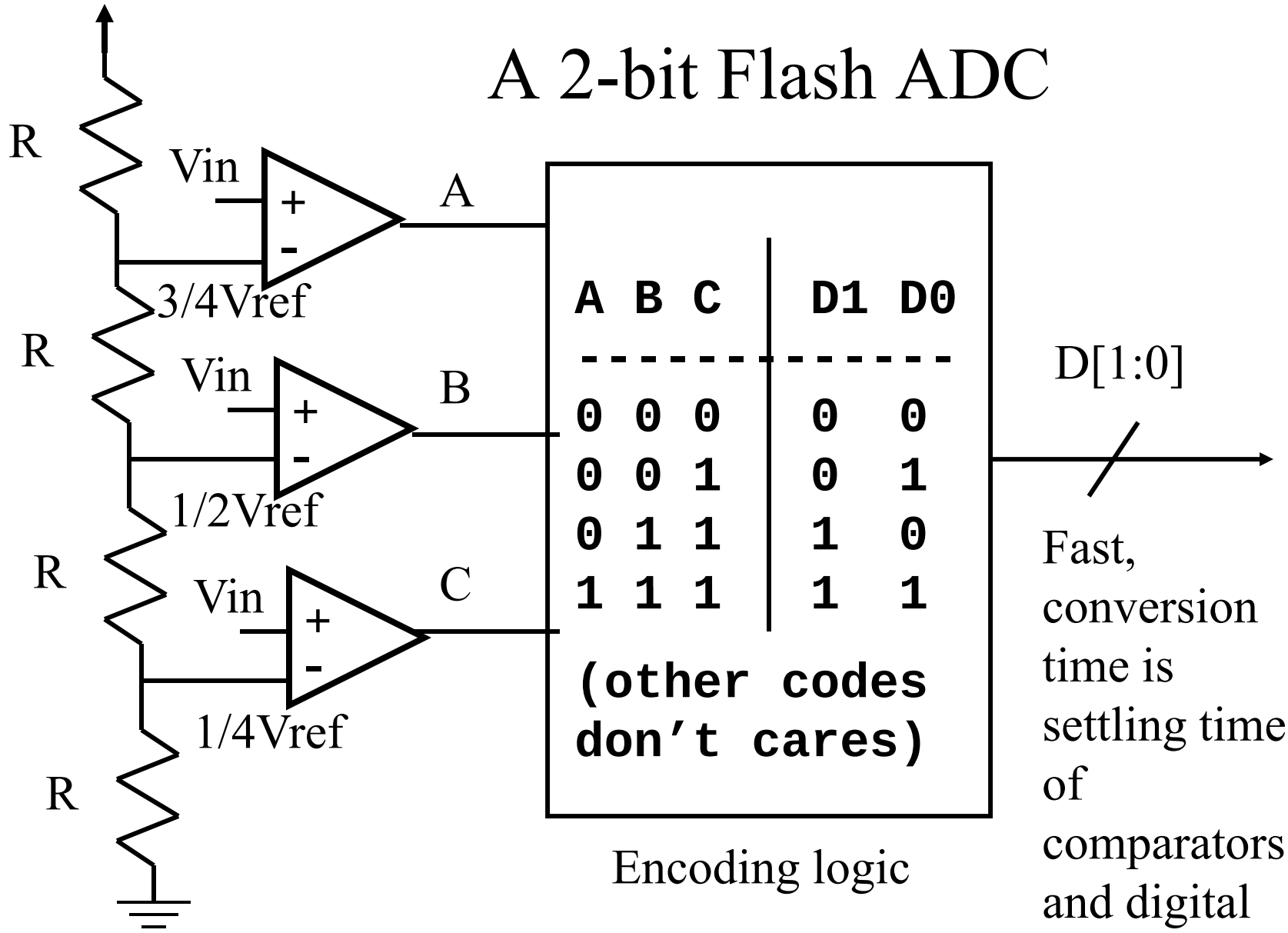
3. Set next bit of DAC to '1', DAC input = 0b1110 = 14, so Vdac = 14/16*4= 3.5V.
   Vdac (3.5 V) > Vin (3.14159 V), so guess of '0' for bit1 of DAC was incorrect.
   Reset this bit to '0'.

4. Set last bit of DAC to '1', DAC input = 0b1101 = 13, so Vdac = 13/16*4 = 3.25V.
   Vdac (3.25 V) > Vin (3.14159 V), so guess of '0' for bit0 of DAC was incorrect.
   Rest this bit to '0'.

Final ADC output code is 0b1100.

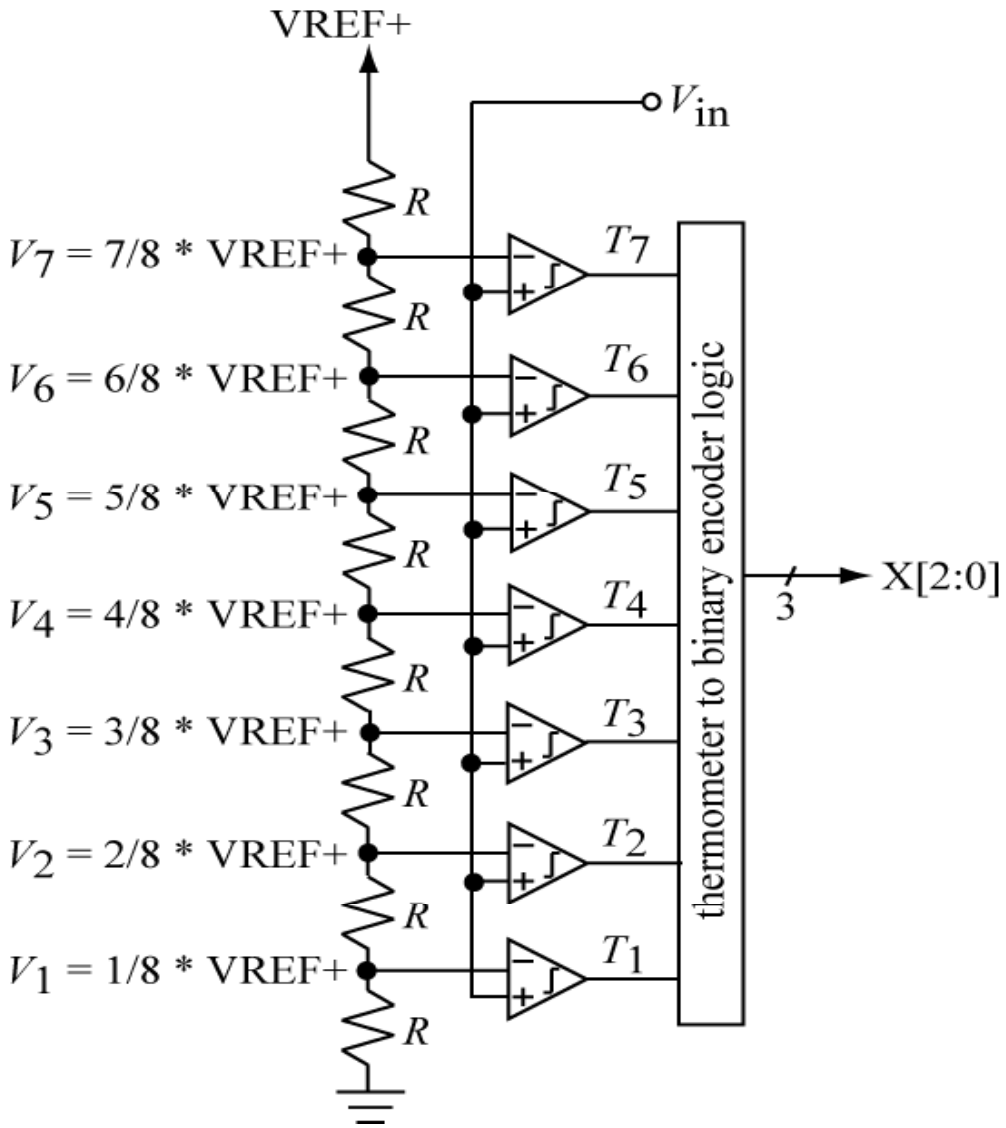
Check result: output code = Vin/Vref * 2^N = 3.14159/4 * 16 = 12.57 = 12 (truncated).

A 2-bit Flash ADC



Fast, conversion time is settling time of comparators and digital logic.

3-bit Flash ADC



ADC Architecture Summary

- ***Flash*** ADCs
  - Fastest possible conversion time
  - Requires the most transistors of any architecture
  - N-bit converter requires 2^N-1 comparators.
  - Commercially available flash converters up to 12 bits.
  - Conversion done in one clock cycle
- ***Successive approximation*** ADCs
  - Use only one comparator
  - Take one clock cycle per bit
  - High precision (16-bit converters are available)

Commercial ADCs

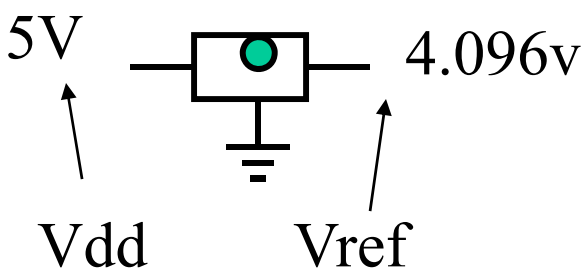
- Key timing parameter is ***conversion time*** – how long does it take to produce a digital output once a conversion is started
- Up to 16-bit ADCs available
- Separated into fast/medium/low speed families
  - Serial interfaces common on medium/low speed ADCs
- For high-precision ADCs, challenge is keeping system noise from affecting conversion
  - Assume a 16-bit DAC, and a 4.1V reference, then 1 LSB $= 4.1/2^{16} =$ 62 μV.

Voltage References

Stability of voltage reference is critical for high precision conversions.

We will use Vdd as our voltage reference for convenience, but will be throwing away at least two bits of precision due to Vdd fluctuations.

Example commercial voltage reference: 2.048v, 2.5v , 3v, 3.3v, 4.096v, 5v (Maxim 6029). The PIC24H can only use a voltage reference of either 3.0 V or 3.3 V.



Key parameter for a voltage is stability over temperature operating range. Need this to be less than ½ of a LSB value.