Analog-to-Digital Conversion

Terminology

analog: continuously valued signal, such as temperature or speed, with infinite possible values in between

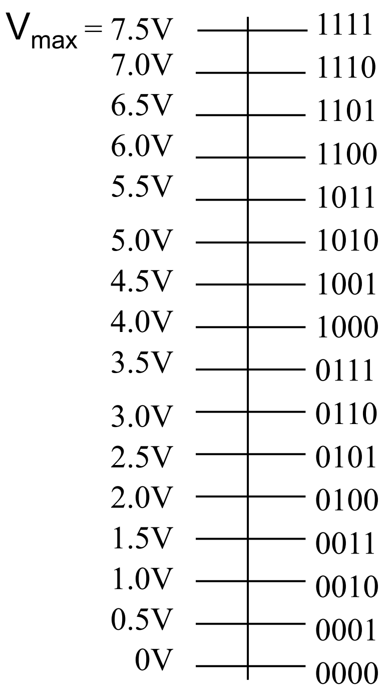
digital: discretely valued signal, such as integers, encoded in binary

analog-to-digital converter: ADC, A/D, A2D; converts an analog signal to a digital signal

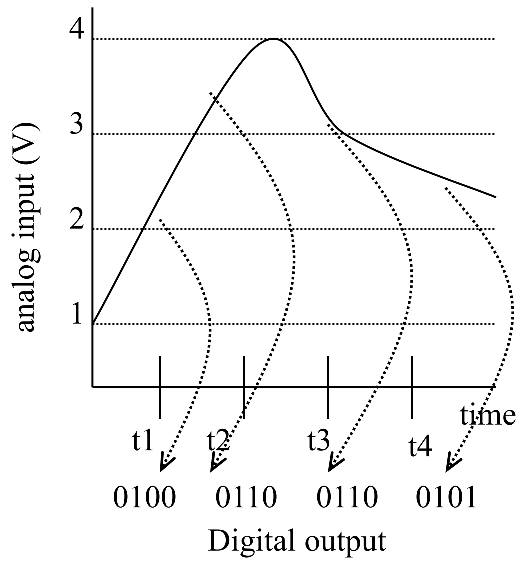
digital-to-analog converter: DAC, D/A, D2A

An embedded system's surroundings typically involve many analog signals.

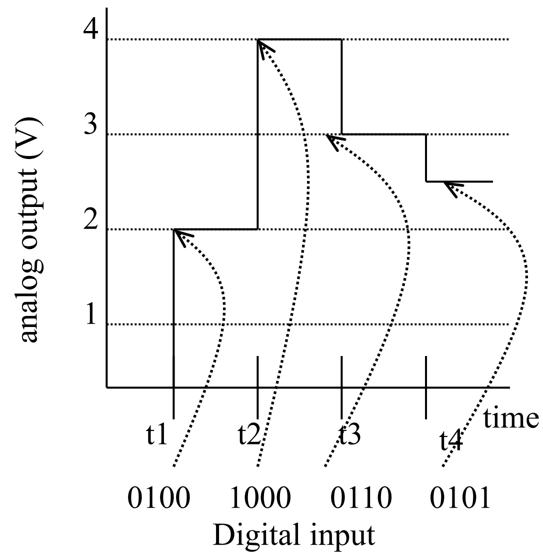
Analog-to-digital converters



proportionality



analog to digital



digital to analog

Proportional Signals

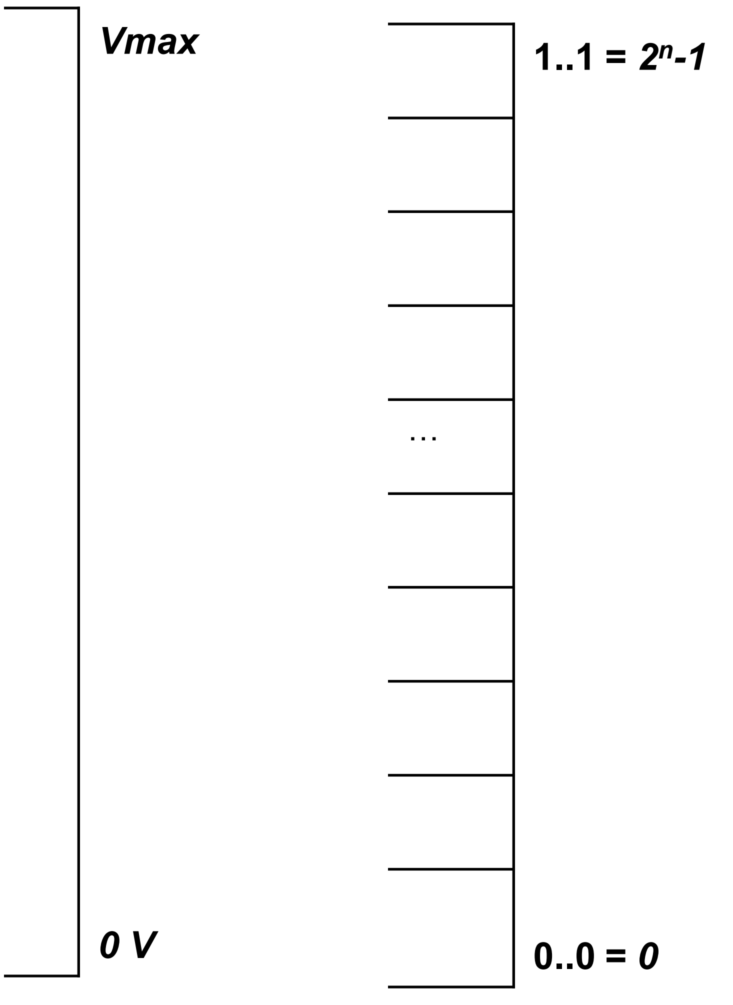
Simple Equation

Assume minimum voltage of 0 V.
Vmax = maximum voltage of the analog signal
a = analog value
n = number of bits for digital encoding

2^n = number of digital codes
M = number of steps, either 2^n or $2^n - 1$

d = digital encoding

$$a / Vmax = d / M$$



Resolution

Let $n = 2$

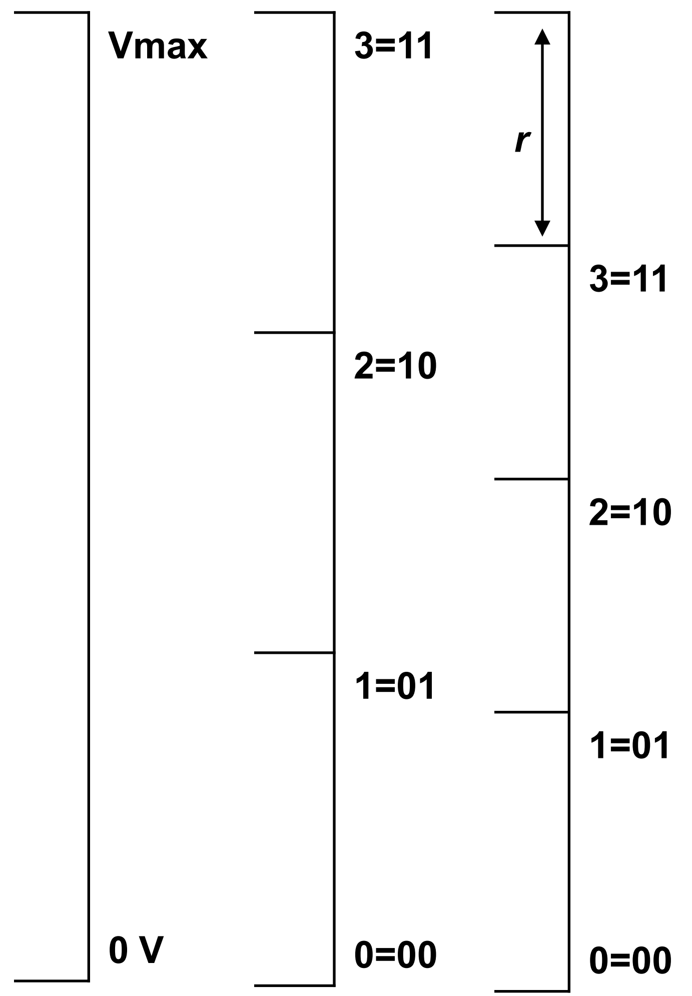
$\underline{M = 2^n - 1}$

3 steps on the digital scale
$d_0 = 0 = 0b00$
$d_{Vmax} = 3 = 0b11$

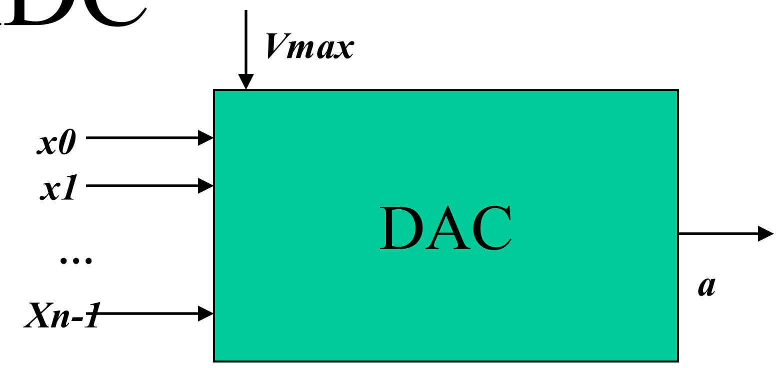
$\underline{M = 2^n}$

4 steps on the digital scale
$d_0 = 0 = 0b00$
$d_{Vmax - r} = 3 = 0b11$ (no d_{Vmax})

r, **resolution**: smallest analog change resulting from changing one bit



DAC vs. ADC



DAC:
n digital inputs for digital encoding *d*
analog input for *Vmax*
analog output *a*

ADC:

Given a *Vmax* analog input and an analog input *a*, how does the converter know what binary value to assign to *d* in order to satisfy the ratio?

- may use DAC to generate analog values for comparison with *a*
- ADC "guesses" an encoding *d*, then checks its guess by inputting *d* into the DAC and comparing the generated analog output *a′* with original analog input *a*
- How does the ADC guess the correct encoding?

ADC: Digital Encoding

Guessing the encoding is similar to finding an item in a list.

1. **Sequential search** – counting up: start with an encoding of 0, then 1, then 2, etc. until find a match.
   - ***2^n*** comparisons: Slow!
2. **Binary search** – successive approximation: start with an encoding for half of maximum; then compare analog result with original analog input; if result is greater (less) than the original, set the new encoding to halfway between this one and the minimum (maximum); continue dividing encoding range in half until the compared voltages are equal
   - ***n*** comparisons: Faster, but more complex converter

→ Takes time to guess the encoding: start conversion input, conversion complete output

ADC using successive approximation

- Given an analog input signal whose voltage should range from 0 to 15 volts, and an 8-bit digital encoding, calculate the correct encoding for 5 volts. Then trace the successive-approximation approach to find the correct encoding.
- Assume $M = 2^n - 1$

$$a / Vmax = d / M$$

$$5 / 15 = d / (256 - 1)$$

$d = 85$ or binary 01010101

ADC using successive approximation

Step 1-4: determine bits 0-3

$\frac{1}{2}(V_{max} - V_{min}) = 7.5$ volts
$V_{max} = 7.5$ volts.

0	0	0	0	0	0	0	0

$\frac{1}{2}(7.5 + 0) = 3.75$ volts
$V_{min} = 3.75$ volts.

0	1	0	0	0	0	0	0

$\frac{1}{2}(7.5 + 3.75) = 5.63$ volts
$V_{max} = 5.63$ volts

0	1	0	0	0	0	0	0

$\frac{1}{2}(5.63 + 3.75) = 4.69$ volts
$V_{min} = 4.69$ volts.

0	1	0	1	0	0	0	0

ADC using successive approximation

Step 5-8: Determine bits 4-7

$\frac{1}{2}(5.63 + 4.69) = 5.16$ volts
$V_{max} = 5.16$ volts.

0	1	0	1	0	0	0	0

$\frac{1}{2}(5.16 + 4.69) = 4.93$ volts
$V_{min} = 4.93$ volts.

0	1	0	1	0	1	0	0

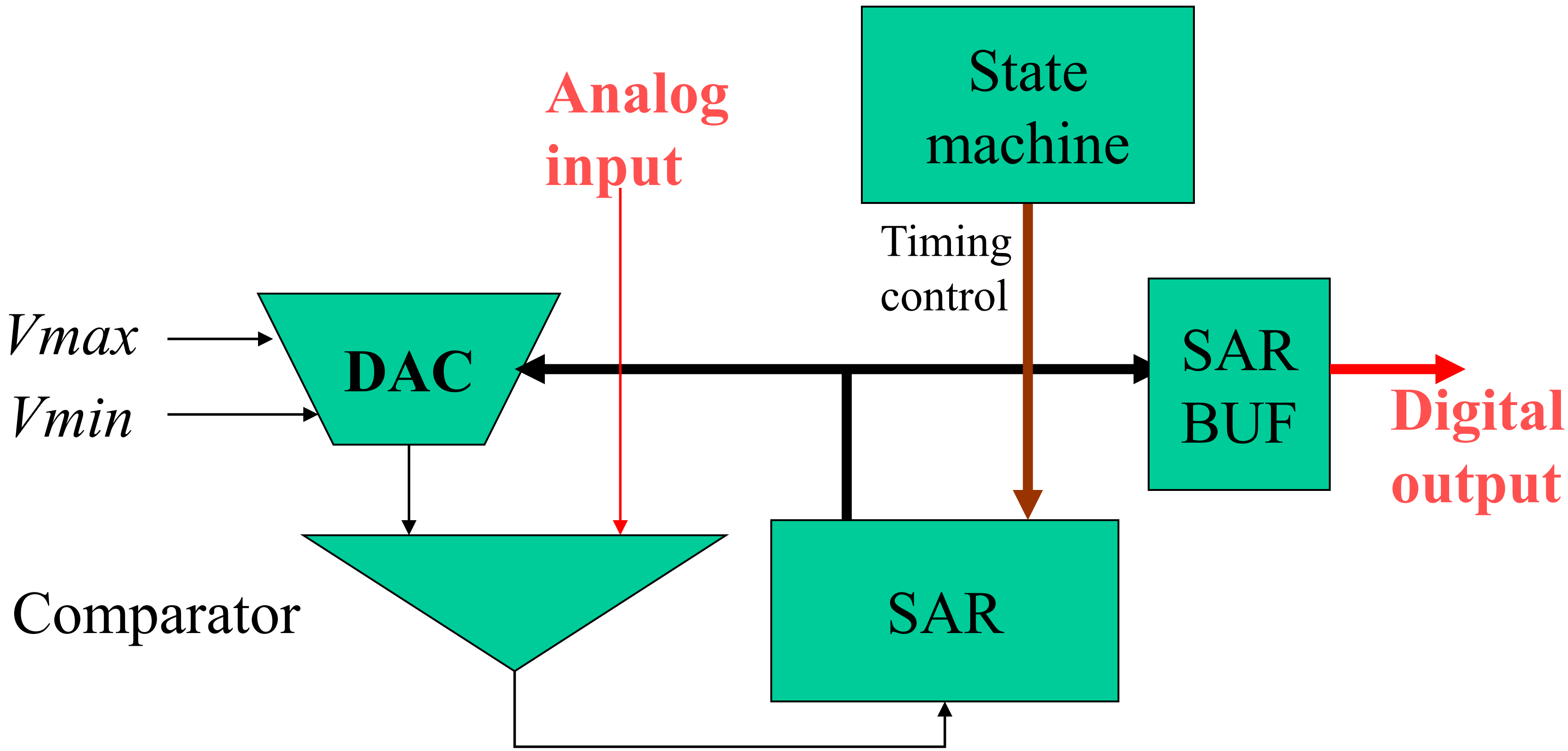
$\frac{1}{2}(5.16 + 4.93) = 5.05$ volts
$V_{max} = 5.05$ volts.

0	1	0	1	0	1	0	0

$\frac{1}{2}(5.05 + 4.93) =$
4.99 volts

0	1	0	1	0	1	0	1

Constructing ADC



SAR: Successive approximation register

Terms & Equations

Offset: minimum analog value

Span (or **Range**): difference between maximum and minimum analog values

Max - Min

n: number of bits in digital code (sometimes referred to as *n*-bit resolution)

Bit Weight: analog value corresponding to a bit in the digital number

Step Size (or **Resolution**): smallest analog change resulting from changing one bit in the digital number, or the analog difference between two consecutive digital numbers; also the bit weight of the

Span / 2^n (Assume $M = 2^n$)

Let AV be Analog Value; DN be Digital Number:

$$AV = DN * \text{Step Size} + \text{Offset} = (DN / 2^n) * \text{Span} + \text{Offset}$$

$$DN = (AV - \text{Offset}) / \text{Step Size} = (AV - \text{Offset}) * 2^n / \text{Span}$$

Scan Sequence and Conversion

- After the ADC is initialized, a sequence of scans is set up as a “queue” in the CCW Table.
- Each channel to be scanned is added to the queue at successive positions 0, 1, 2, etc. For example: CCW0, CCW1, CCW2, CCW3.
  - An end-of-queue marker should be added at the next position.
- The ADC starts the scan and conversion when it is triggered by the enable bit.
- The ADC reads the CCWs, one after another until end-of-queue is reached, and for each CCW, it converts the signal on the specified channel.
  - A conversion on a channel stores a result in the respective position of the Result Table, e.g., the result for CCW0 is stored at Result0, etc.
- When the scan and conversion is complete for all CCWs, then the ADC sets the completion flag to 1. Now all digital results are available to be read from the Result Table.