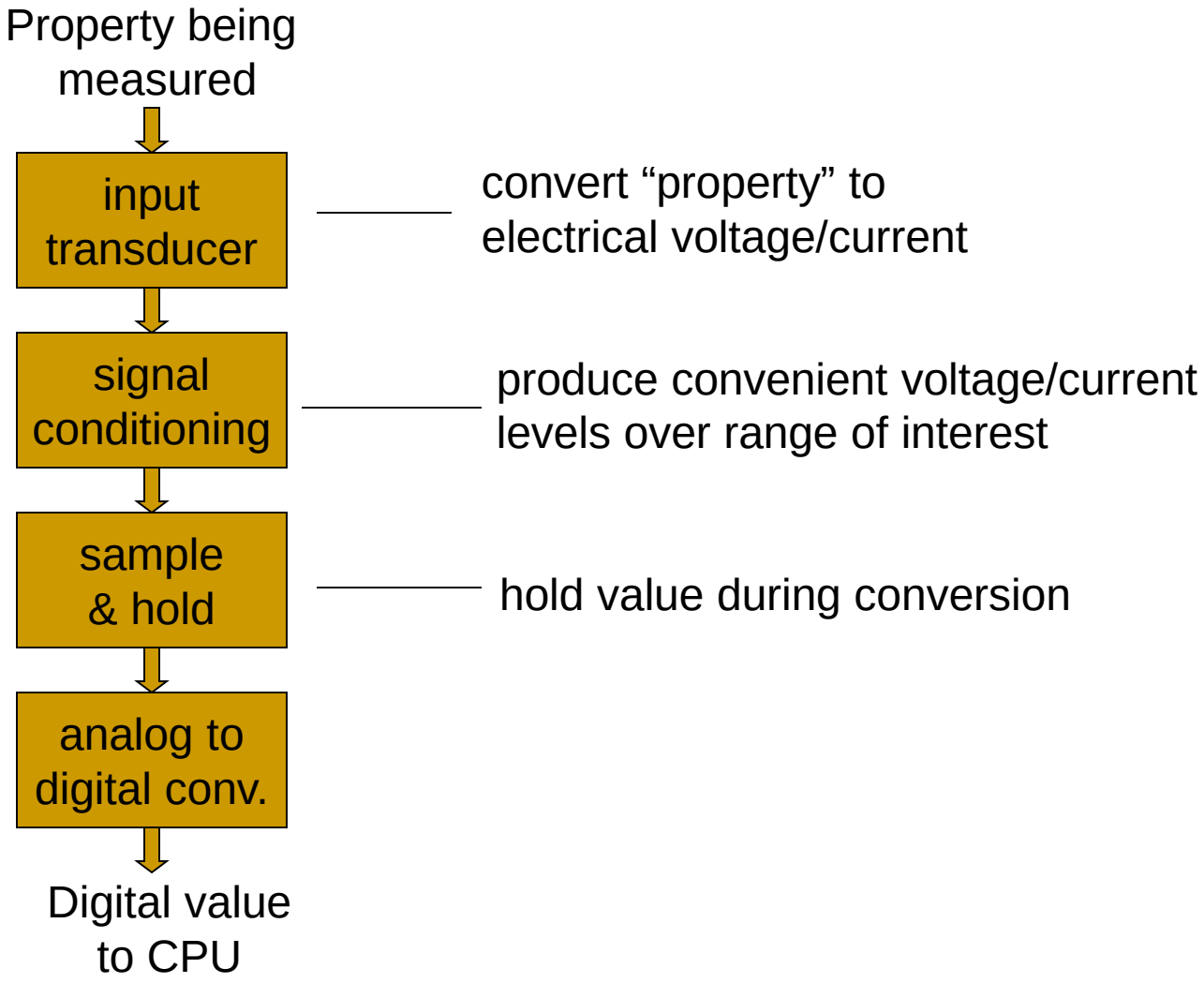
Analog Input/Output Subsystem Design

Reference:

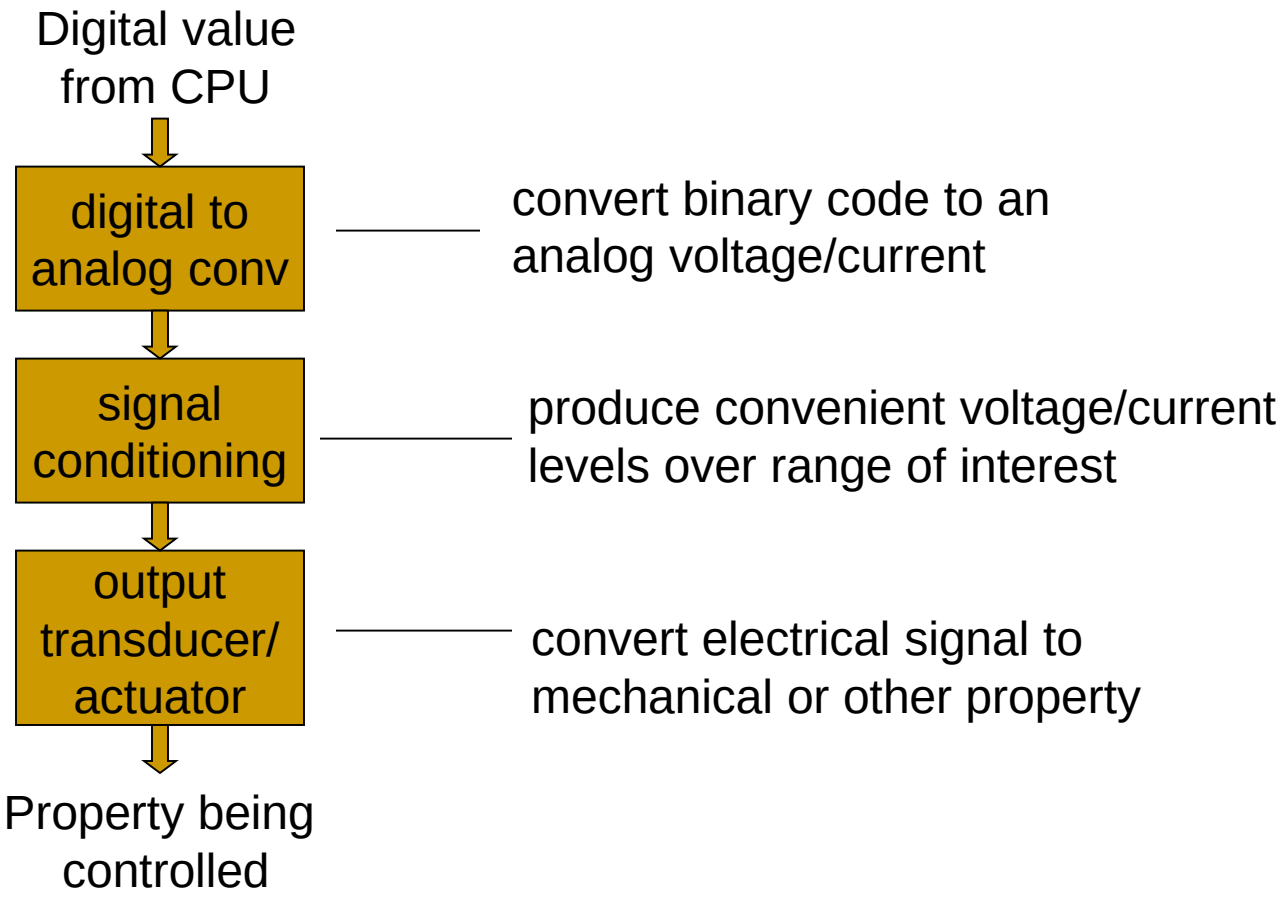
STM32F4xx Reference Manual

(ADC, DAC chapters)

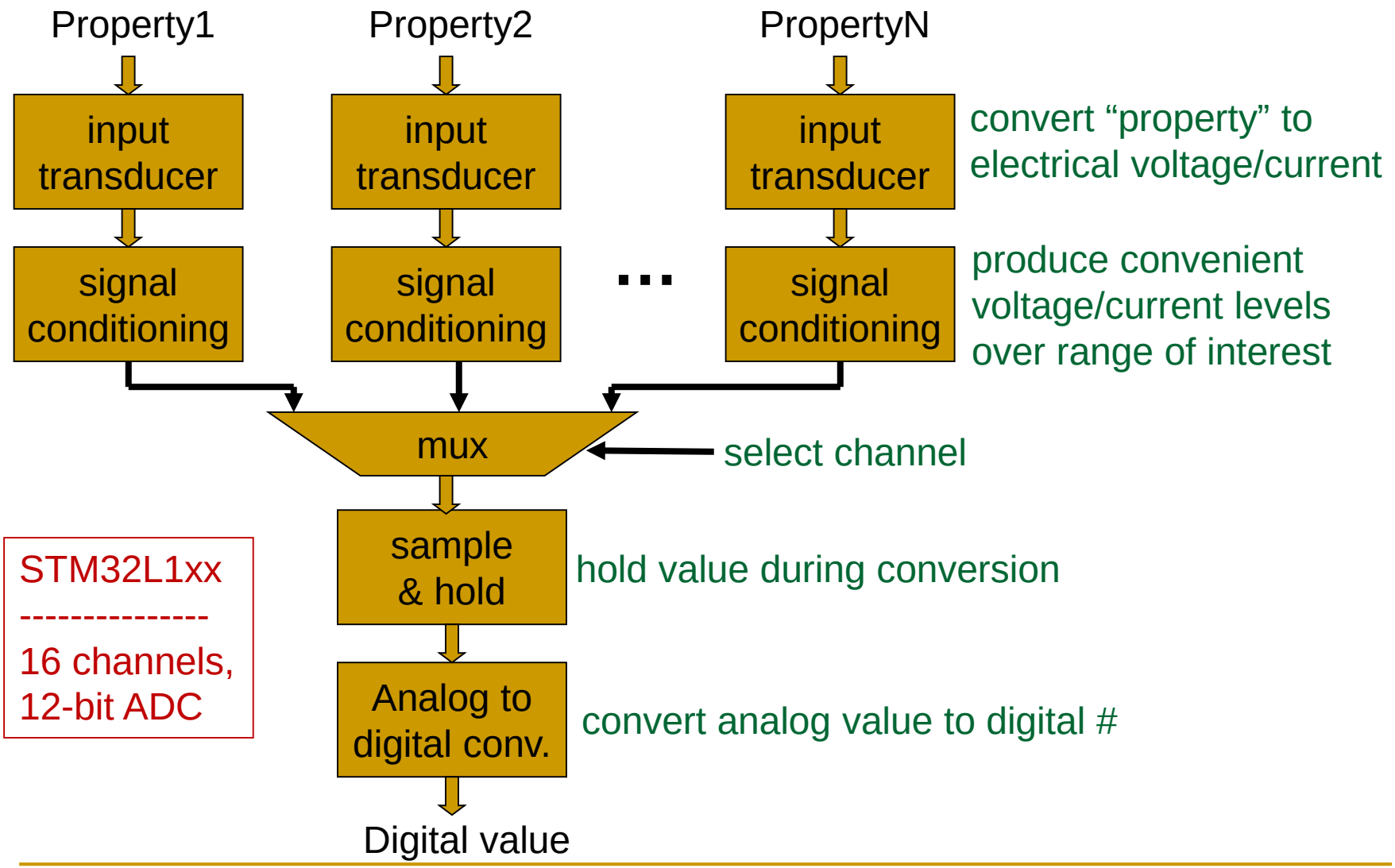
Analog input subsystem



Analog output subsystem



Typical analog input subsystem

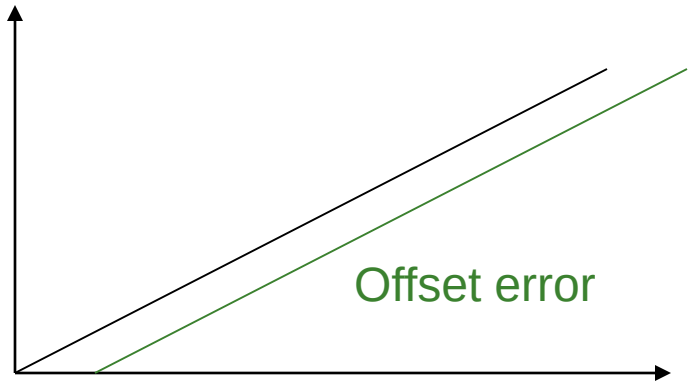


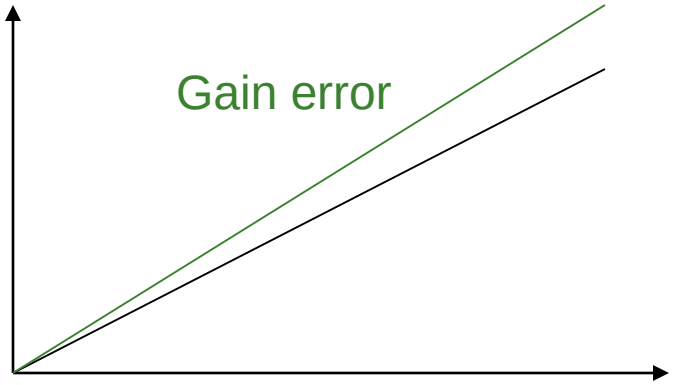
Analog subsystem properties

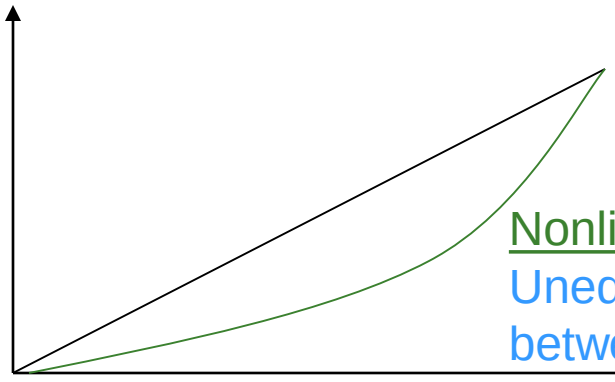
- Accuracy: degree to which measured value differs from true value
- Resolution/precision: degree to which two conditions can be distinguished
  - Related to #bits in digital value
- Range: minimum to maximum "useful" value
- Linearity: $y = Ax + B$ (correction req'd if not linear)
  - piecewise linear approximation over different ranges
- Repeatability: same measurement for a given value
  - affected by hysteresis or other phenomena
- Stability: value changes other than due to the property being measured (eg. T affecting P)

Analog to digital conversion errors

May need to correct in software

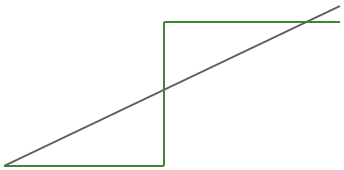






Nonlinearity error -
Unequal distances
between transition points

Quantization error:
Difference between digital & analog values
Usually want ± ½ LSB

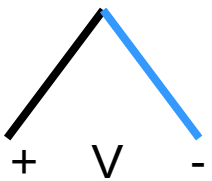


Transducers

- Convert physical quantity to electrical signal
  - Self-generating – generates voltage/current signal
  - Non-self-generating – other property change (ex. R)
- Examples:
  - Force/stress (strain gage)
  - Temperature (thermocouple, thermistor, semicond.)
  - Pressure
  - Humidity (gypsum block)
  - Smoke
  - Light (phototransistor, photoconductive cell)
  - Acceleration (accelerometer)
  - Flow
  - Position (potentiometer, displacement)

Temperature sensors

- Thermocouple – “Seeback EMF produced by heating junction of dissimilar metals (μV)



- Thermistor – mix of materials in ceramic

$$R_t = R_0 e^{\beta[1/T - 1/To]}$$

  - Negative temperature coefficient: R^ with Tv
  - Linear over small range

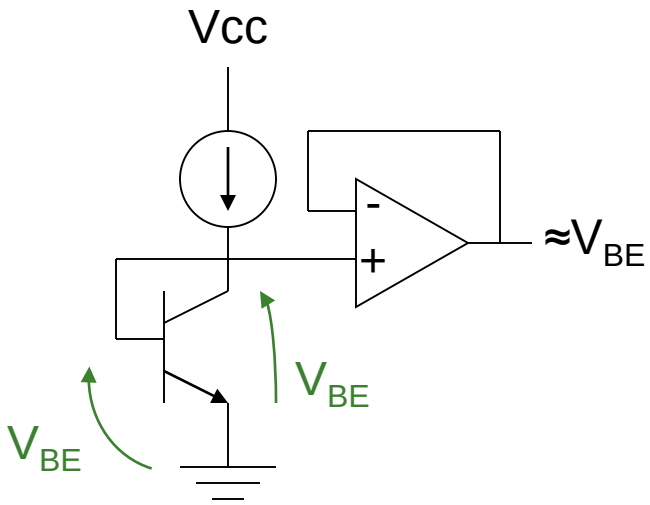
- Metal conductor:

$$R_t = R_0[1 + \alpha(T - T_0)]$$

  - Positive temp. coefficient: R^ with T^

Semiconductor temperature sensor

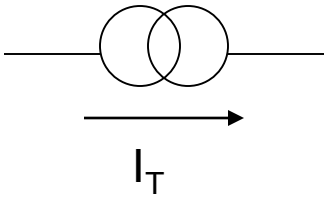
Base-emitter voltage approximately proportional to T



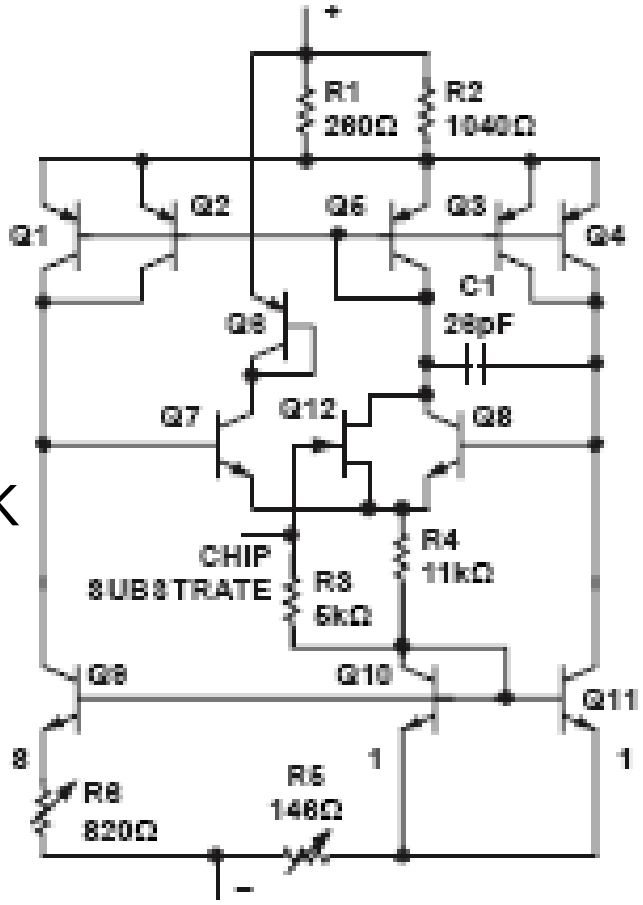
$$V_{BE} = \frac{kT}{q} \ln\left[\frac{Ic}{Is}\right]$$

$$V_{BE} \propto T$$

Analog Devices AD590 Temperature Transducer



- IC generates current proportional to temperature
- Generated current I_T is linear: 1 μa/^{o}K



Example:

Design a temperature monitor with output in the range [0v..4v] over temperature range [-20^{o}C .. +60^{o}C]

(Use summing amplifier)

Strain Gage

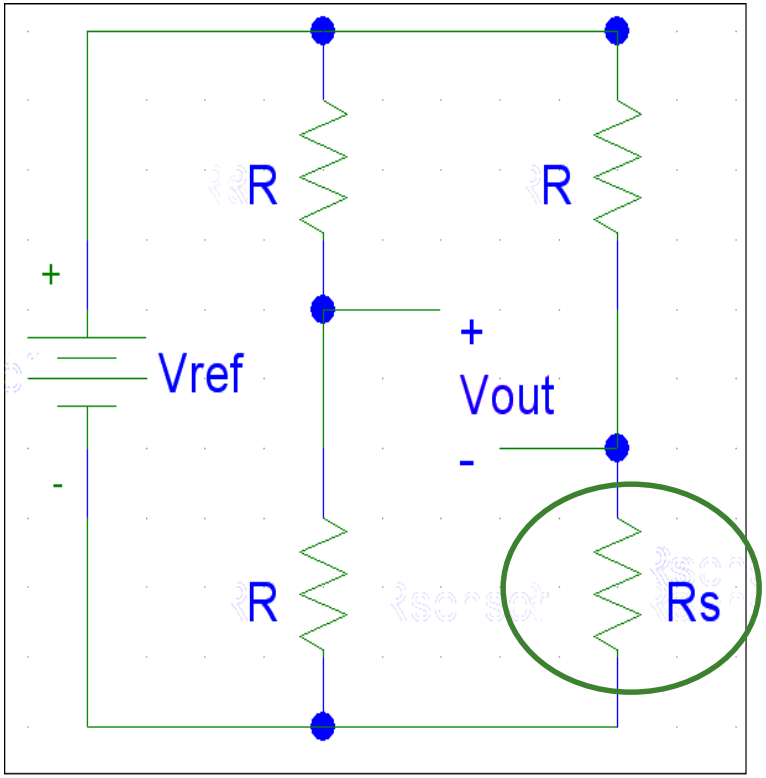
- Measure stress by measuring change or resistance of a conductor due to change of its length/area

$$R = \rho\left(\frac{Lo}{Ao}\right)$$

- Compression: L decreases, A increases
- Elongation: L increases, A decreases
- "Gage factor" (sensitivity): $S = \dfrac{\Delta R / R}{\Delta L / L}$

Wheatsone bridge

- Measure small resistance changes



$$Vo = V_{ref}\left(\frac{R}{R+R}\right) - V_{ref}\left(\frac{Rs}{R+Rs}\right)$$

$$= V_{ref}\left[\frac{1}{2} - \frac{Rs}{R+Rs}\right]$$

"Balanced": Vo = 0 when R=Rs

Some pressure sensors use bridge with all 4 R's variable

Signal conditioning

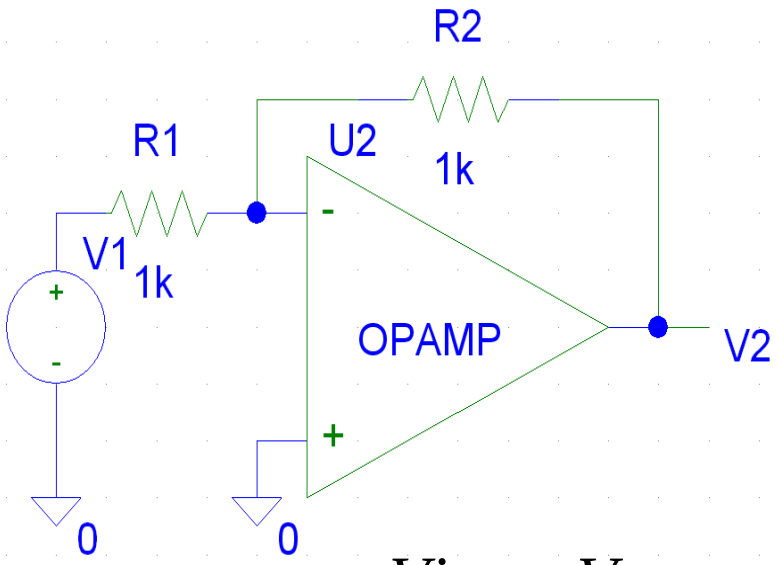
- Produce noise-free signal over “working” input range
  - Amplify voltage/current levels
  - Bias (move levels to desired range)
  - Filter to remove noise
  - Isolation/protection (optical/transformer)
  - Common mode rejection for differential signals
  - Convert current source to voltage
- Conditioning often done with op amp circuits

Operational amplifiers

- Amplifier types:
  - Inverting amplifier
  - Non-inverting amplifier
  - Summing amplifier
  - Differential amplifier
  - Instrumentation amplifier
- Tradeoffs
  - Inverting/noninverting
  - High input impedance
  - Defined gain
  - Comon mode rejection

Basic op amp configurations

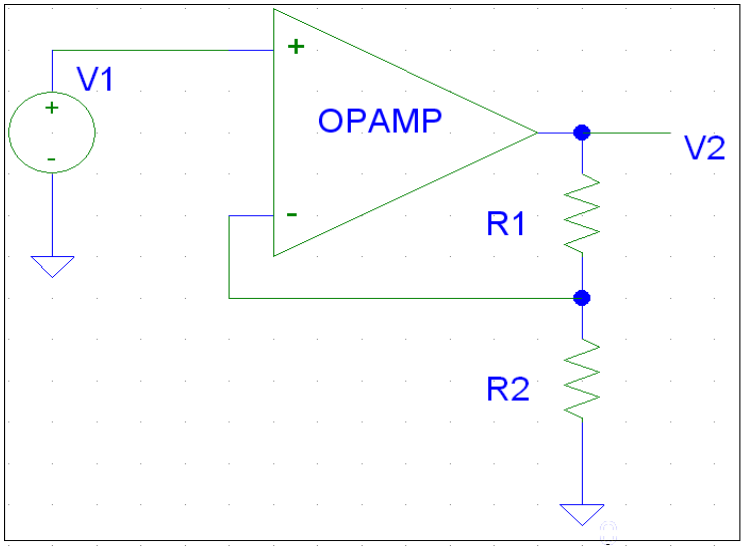
Inverting amplifier



$$\frac{Vi}{R1} = -\frac{Vo}{R2}$$

$$\frac{Vo}{Vi} = -\frac{R2}{R1}$$

Noninverting amplifier

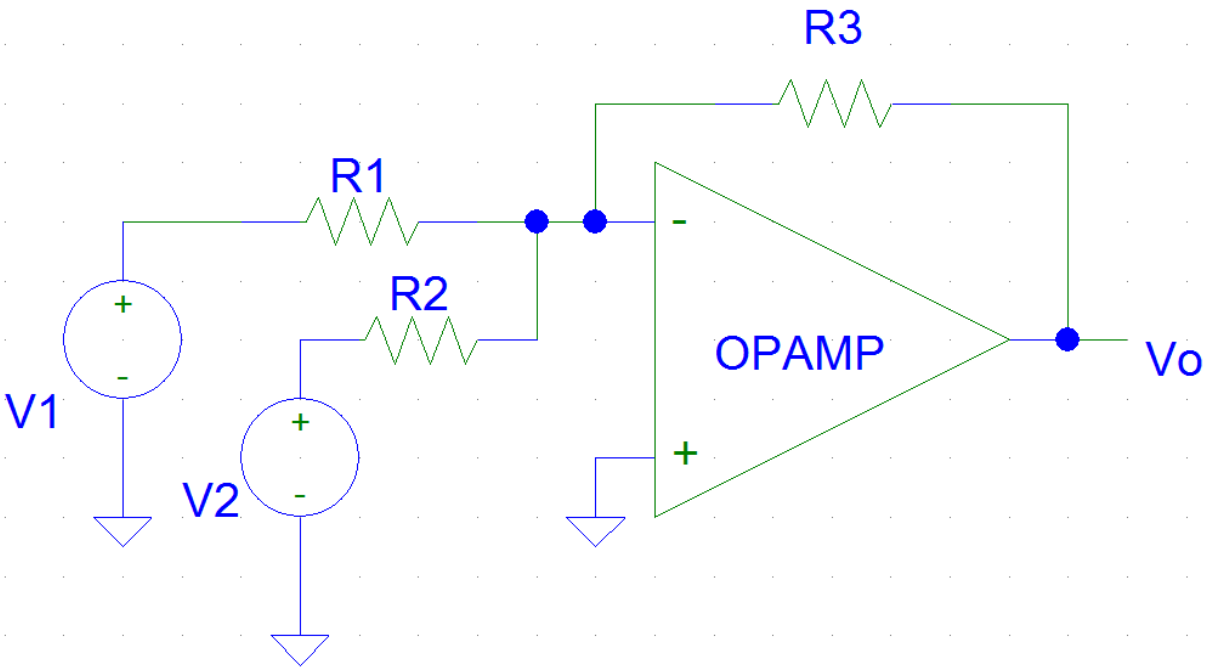


$$Vi = Vo\left(\frac{R2}{R1+R2}\right)$$

$$Vo/Vi = \frac{R1+R2}{R2}$$

Noninverting version has high input impedance

Summing amplifier



$$\frac{V1}{R1}+\frac{V2}{R2}=-\frac{Vo}{R3}$$

$$Vo=-R3\left(\frac{V1}{R1}+\frac{V2}{R2}\right)$$

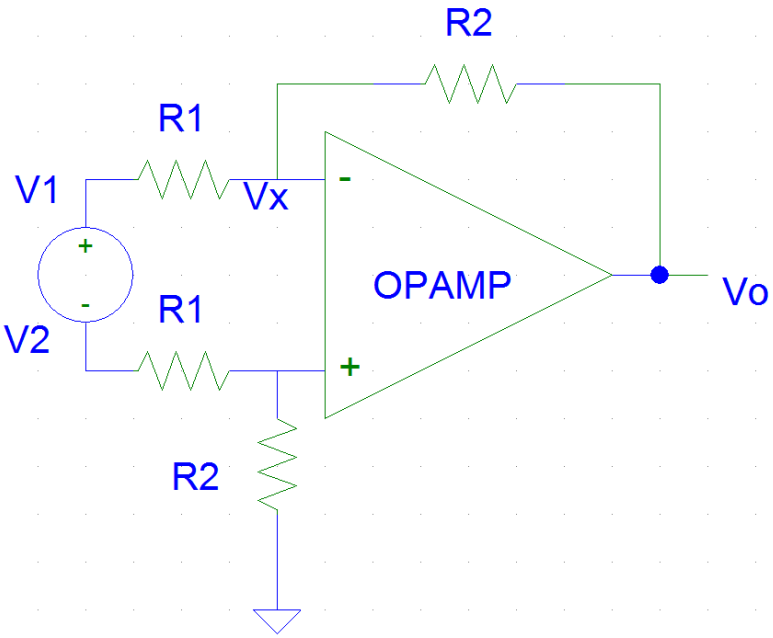
<u>Potential application:</u>
V1 = input voltage
V2/R2 provide an “offset” to V1/R1
(ex. to produce Vo=0 at some V1 value)

Differential amplifier

Eliminates “common mode” voltage (noise, etc.)

$$\frac{V1-Vx}{R1}=\frac{Vx-Vo}{R2} \qquad \frac{V2-Vx}{R1}=\frac{Vx}{R2}$$

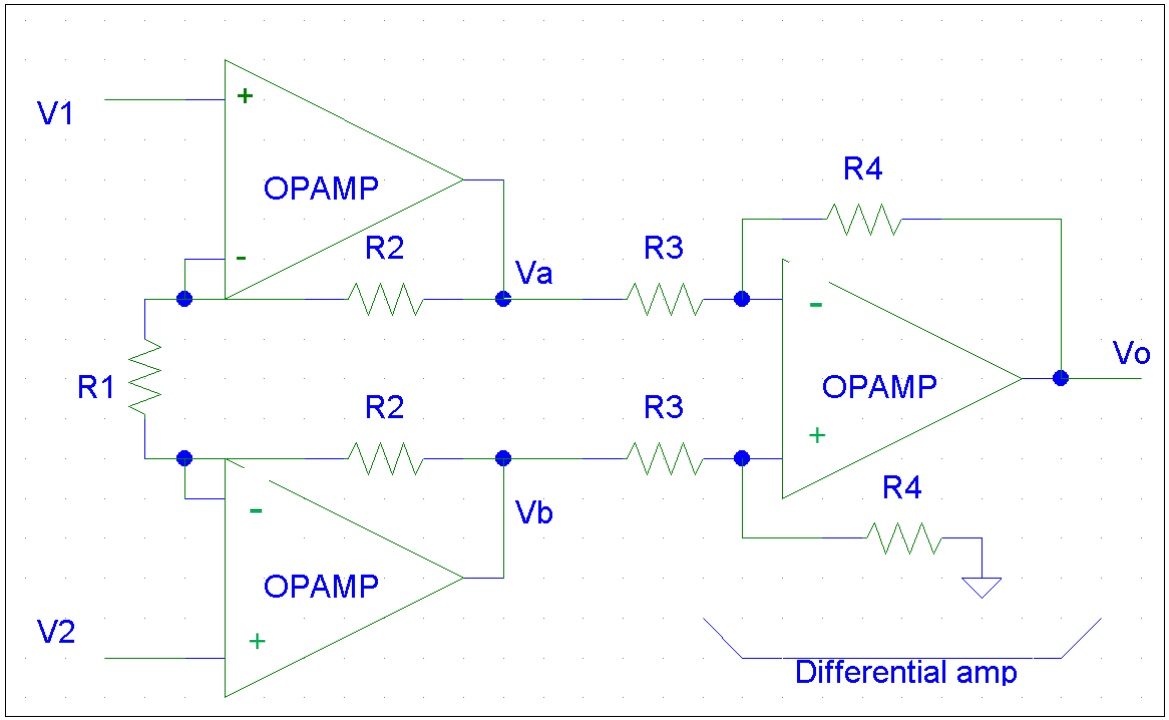
$$Vx=\frac{R1V0+R2V1}{R1+R2} \qquad Vx=\frac{V2R2}{R1+R2}$$



$$Vo=\frac{R2}{R1}(V2-V1)$$

Choose R1 to set input impedance; R2 to set gain

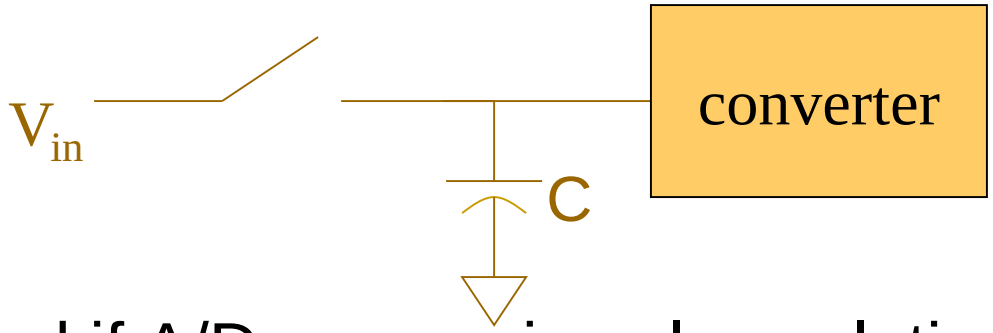
Instrumentation amplifier



$$Vo = (V2 - V1)\left[1 + \frac{2R2}{R1}\right]\left(\frac{R4}{R3}\right)$$

- High input impedance, common mode rejection
- Can match R2, R3, R4 on chip and use external R1 to set gain

Sample-and-hold



- Required if A/D conversion slow relative to frequency of signal:
  - Close switch to “sample” Vin (charge C to Vin)
    - Aperture (sampling) time = duration of switch closure
  - Open switch to “hold” Vin

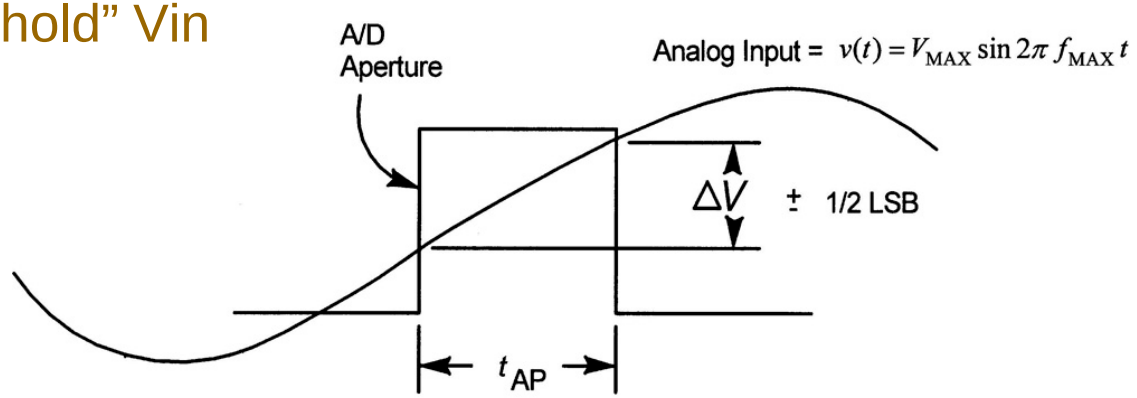


Figure 17-4 Aperture time error.

Analog to digital conversion

- Given: continuous-time electrical signal

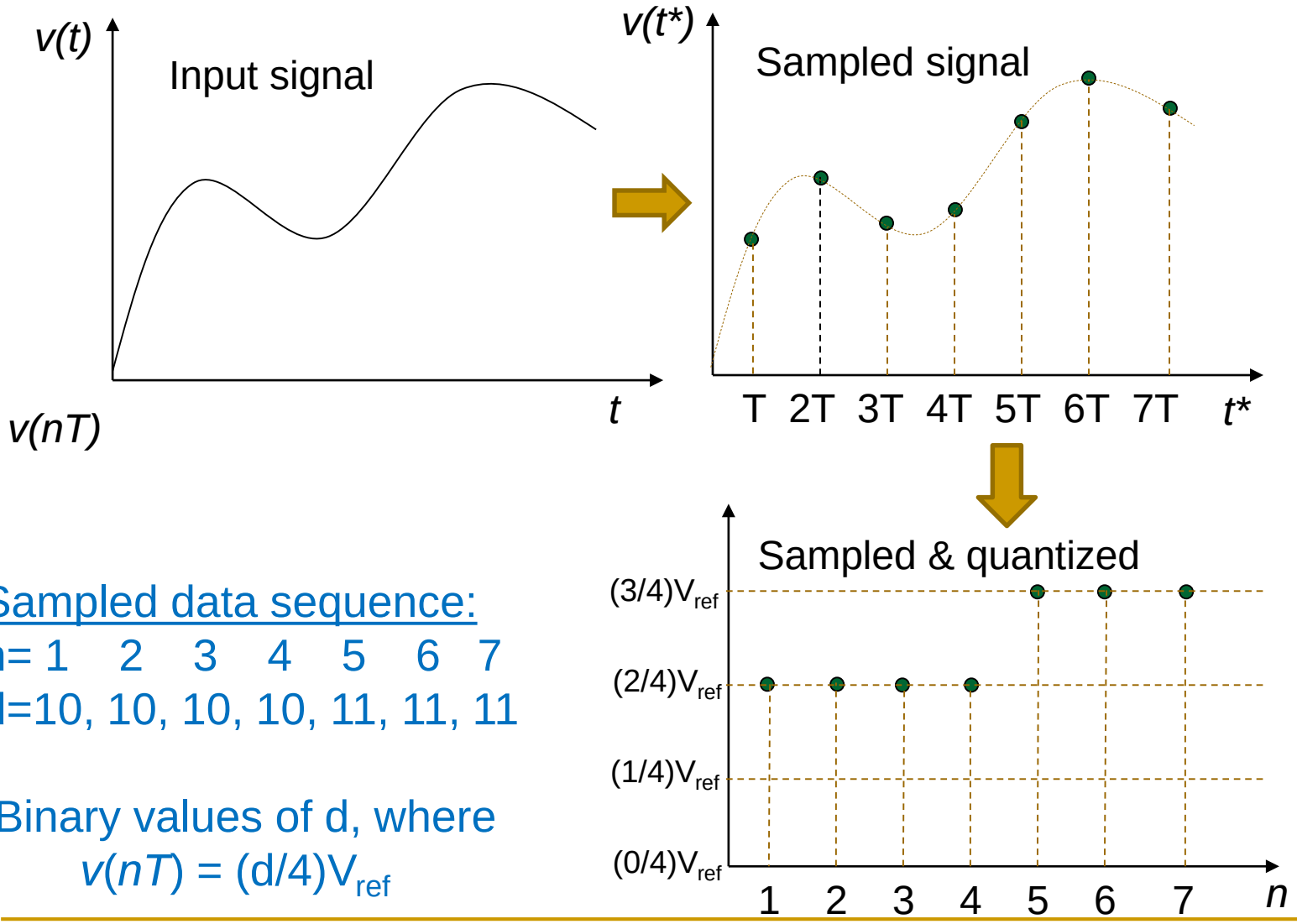
  $v(t), t >= 0$

- Desired: sequence of discrete numeric values that represent the signal at selected sampling times :

  $v(0), v(T), v(2T), \ldots v(nT)$

  - T = "sampling time": v(t) "sampled" every T seconds
  - n = sample number
  - v(nT) = value of v(t) measured at the n^{th} sample time and quantized to one of 2^k discrete levels

A/D conversion process



Sampled data sequence:

n= 1 2 3 4 5 6 7

d=10, 10, 10, 10, 11, 11, 11

Binary values of d, where

$v(nT) = (d/4)V_{ref}$

A/D conversion parameters

- Sampling rate, F (sampling interval T = 1/F)
  - *Nyquist rate* ≥ 2 x (highest frequency in the signal)
    - to reproduce sampled signals
    - CD-quality music sampled at 44.1KHz
      (ear can hear up to about 20-22KHz)
    - Voice in digital telephone sampled at 8KHz

- Precision (# bits in sample value)
  - k = # of bits used to represent sample values
  - "precision": each step represents $(1/2^k) \times V_{range}$
    - Ex. Temperatures [-20^{O}C…+60^{O}C]: if k=8, precision = 80^{O}C/256 = 0.3125^{O}C
  - "accuracy": degree to which converter discerns proper level
    (error when rounding to nearest level)

Analog to digital conversion

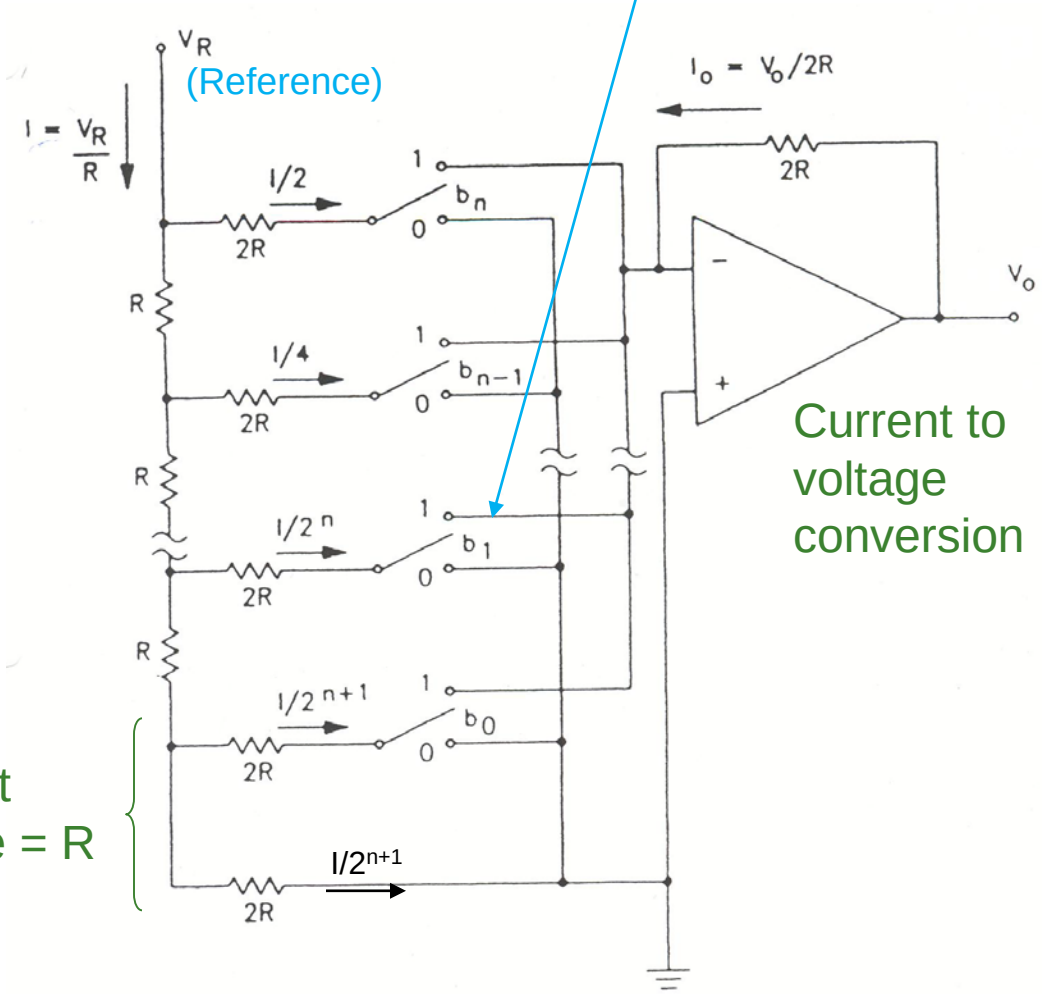
- More difficult than D/A conversion
- Tradeoffs:
  - Precision (# bits)
  - Accuracy
  - Speed (of conversion)
  - Linearity
  - Unipolar vs. bipolar input
  - Encoding method for output
  - Cost
- Often built around digital to analog converters

Digital to analog conversion

Number = $b_n b_{n-1} \ldots b_1 b_0 = b_n*2^n + b_{n-1}*2^{n-1} + \ldots + b_1*2^1 + b_0*2^0$

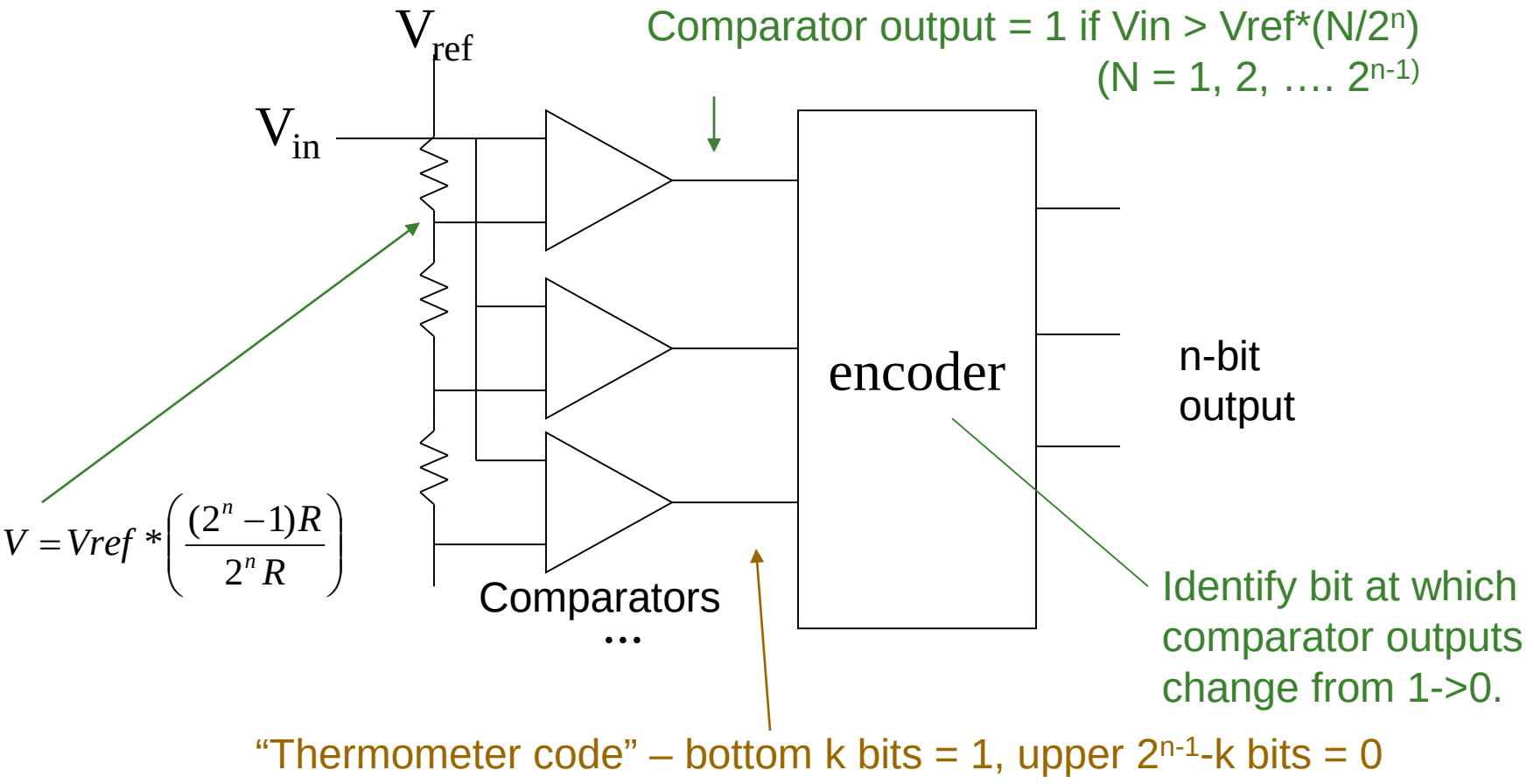
R-2R Ladder Network

$$V_o = V_R \sum_{k=0}^{n} b_k \left(\frac{1}{2^k}\right)$$



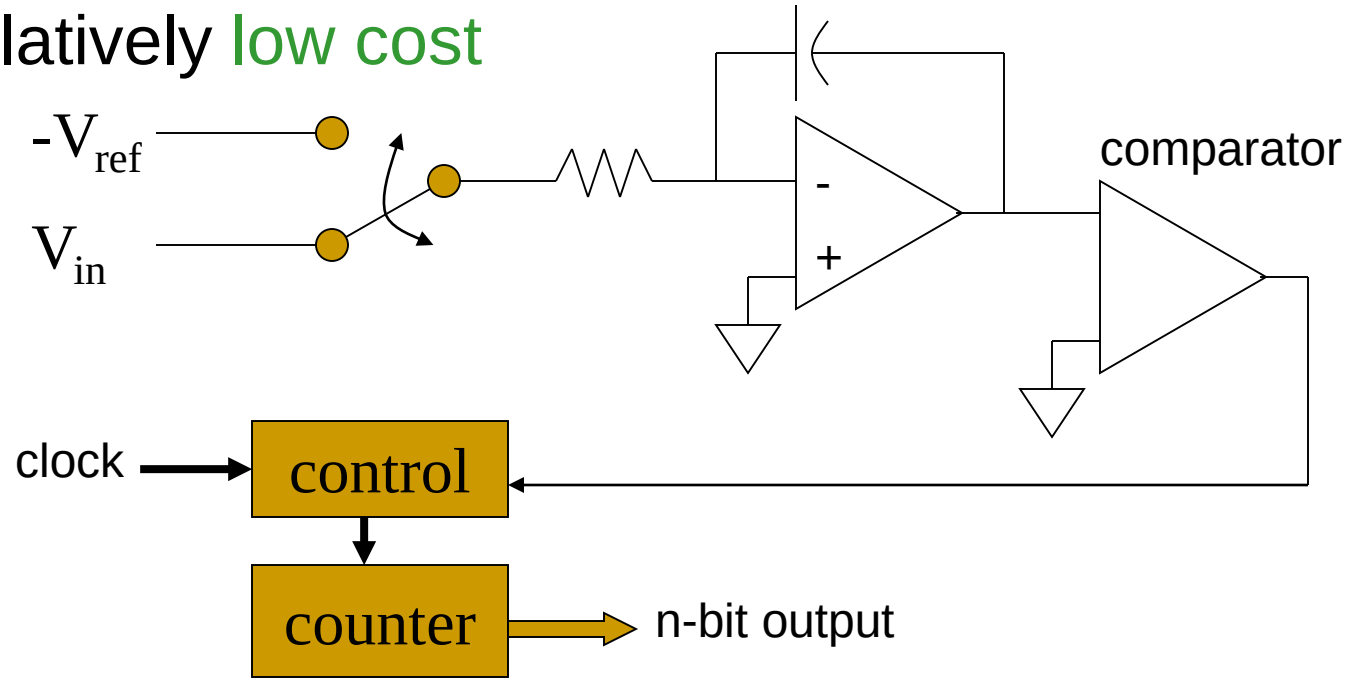
Flash A/D conversion

- N-bit result requires 2^n comparators and resistors:



Dual-slope conversion

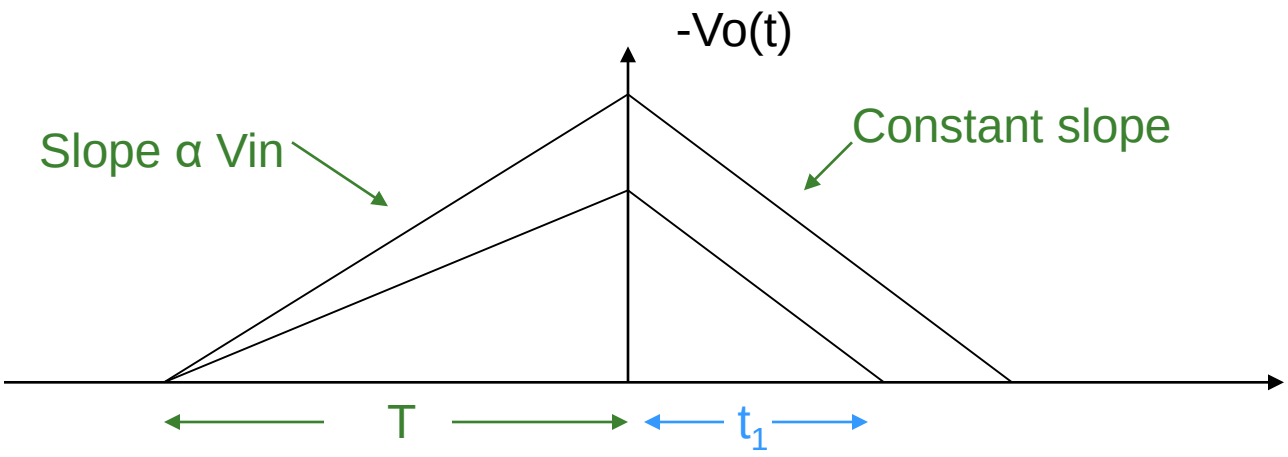
- Use counter to measure time required to charge/discharge capacitor (relatively low speed).
- Charging, then discharging eliminates non-linearities (high accuracy).
- Relatively low cost



Dual-slope conversion steps

1. SW1 connects Vin for fixed time T
   - C charges with current = Vin(t)/R

$$Vo(t) = -\frac{1}{C}\int_0^T i_c(t)dt = -\frac{1}{RC}\int_0^T Vin(t)dt = -\frac{T}{RC}Vin$$



Dual-slope conversion steps

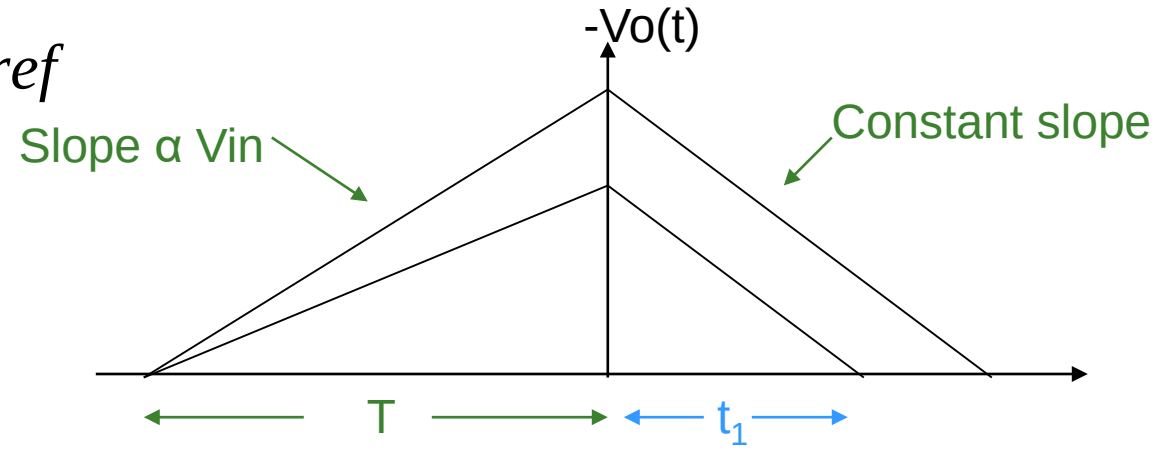
2. SW1 connects –Vref until Vo discharges to 0.
   - C discharges with constant current = -Vref/R

$$Vo(T+t_1) = -\frac{1}{RC}\int_0^T Vin(t)dt + \frac{1}{RC}\int_T^{T+t_1} V_{ref}\,dt$$

   - When Vo(T+t1) = 0:

$$\frac{1}{RC}\int_0^T Vin(t)dt = \frac{1}{RC}\int_T^{T+t_1} V_{ref}\,dt$$

$$Vin = \left(\frac{t_1}{T}\right) Vref$$



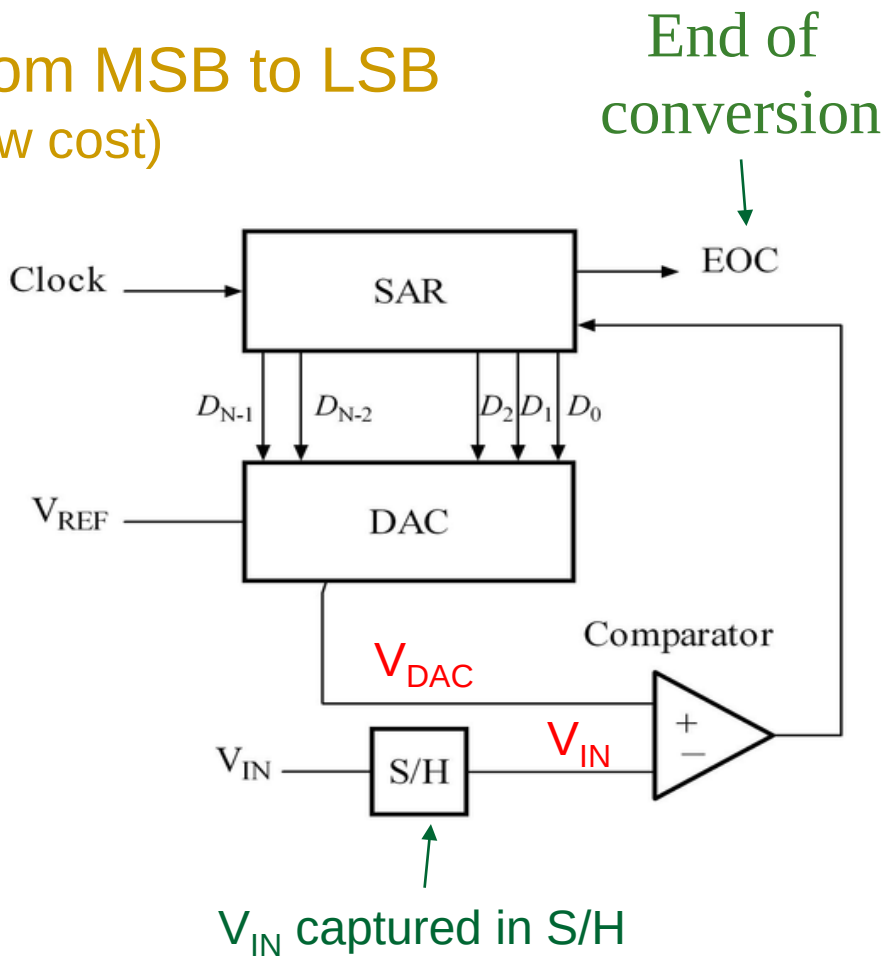
Use a counter to measure t1.

Successive approximation analog to digital converter (ADC)

- Determine one bit at a time, from MSB to LSB
  Used in most microcontrollers (low cost)

1. *Successive Approximation Register* (SAR) sets $D_{N-1} = 1$
2. *SAR* outputs $D_{N-1} \ldots D_0$, converted by *DAC* to analog V_{DAC}
3. V_{DAC} is compared to V_{IN}
4. *Comparator* output resets D_{N-1} to 0 in SAR if $V_{DAC} < V_{IN}$
5. Repeat 1-4 for $D_{N-2} \ldots D_0$ (one clock period per bit)

- Final SAR value $D_{N-1} \ldots D_0$ is digital representation of V_{IN}



Sigma Delta ADC

- High resolution (16 or more bits)
- High integration
- Reasonable cost
- Often used to sample CD-quality audio
  - 16-bit resolution @ 44.1Ksamples/sec
- Oversampling used to spread noise over wider frequency range
- Digital filtering eliminates the noise
  - Gives good dynamic range with simple ADC

Sigma-Delta A/D Converter

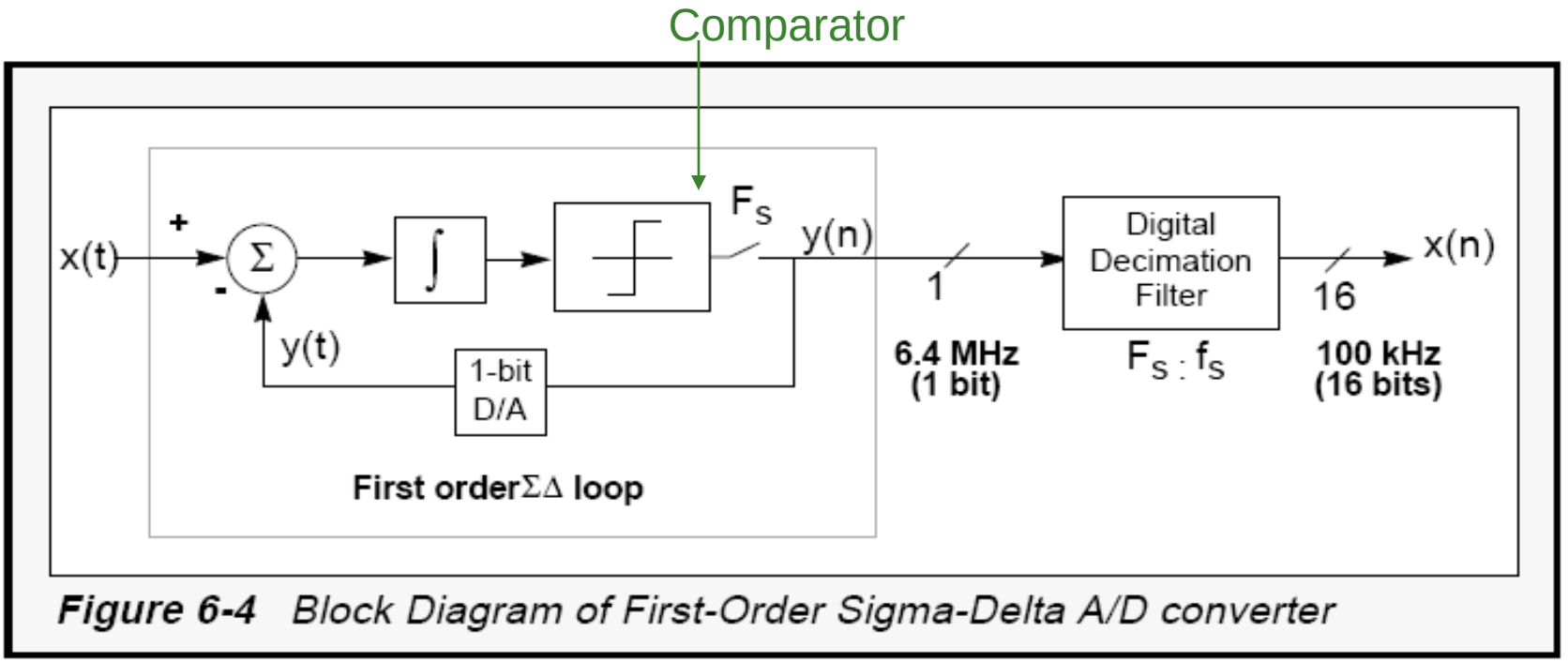
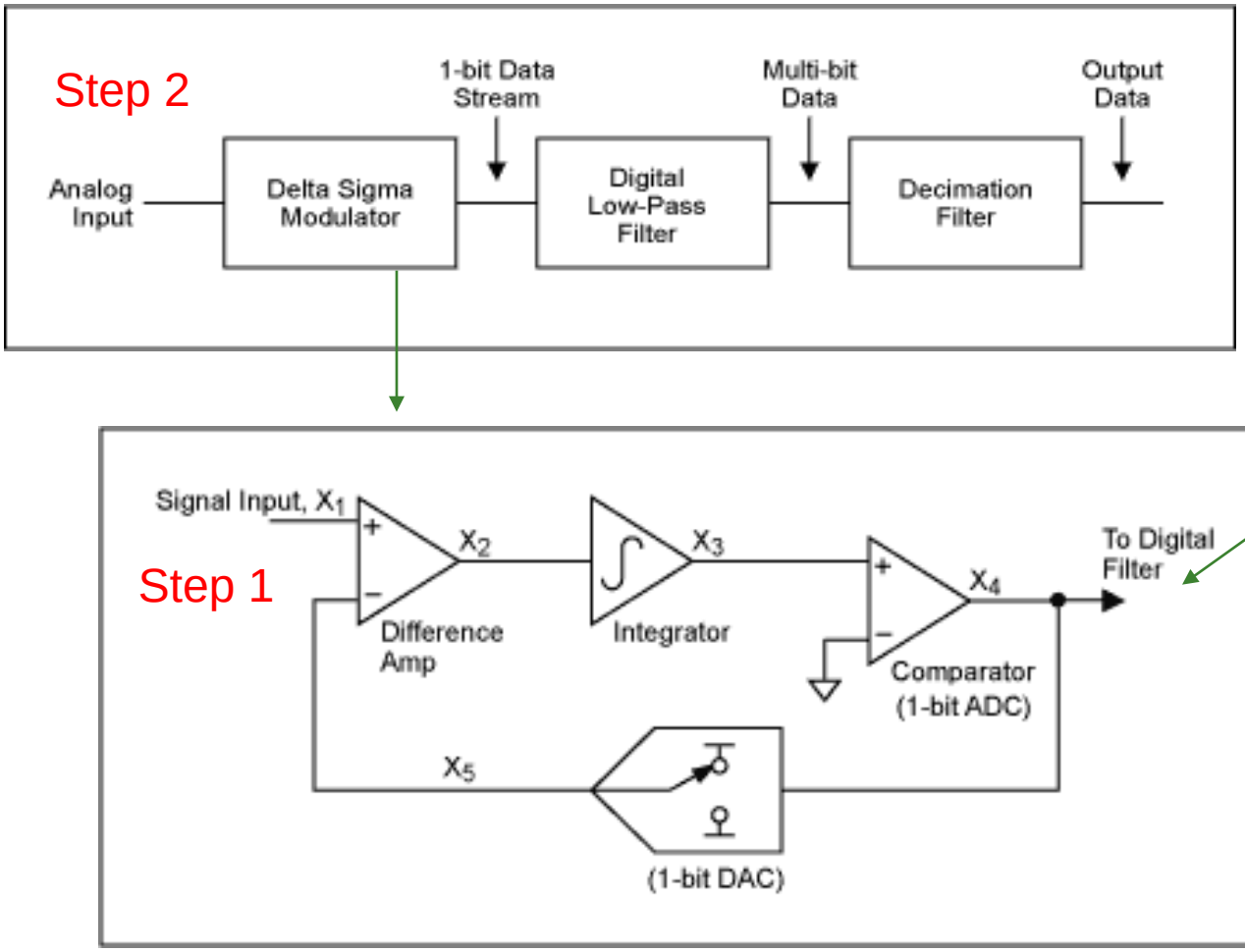


Figure 6-4 *Block Diagram of First-Order Sigma-Delta A/D converter*

Sigma-Delta ADC



Filtering extracts Info from serial data stream. (lower rate)

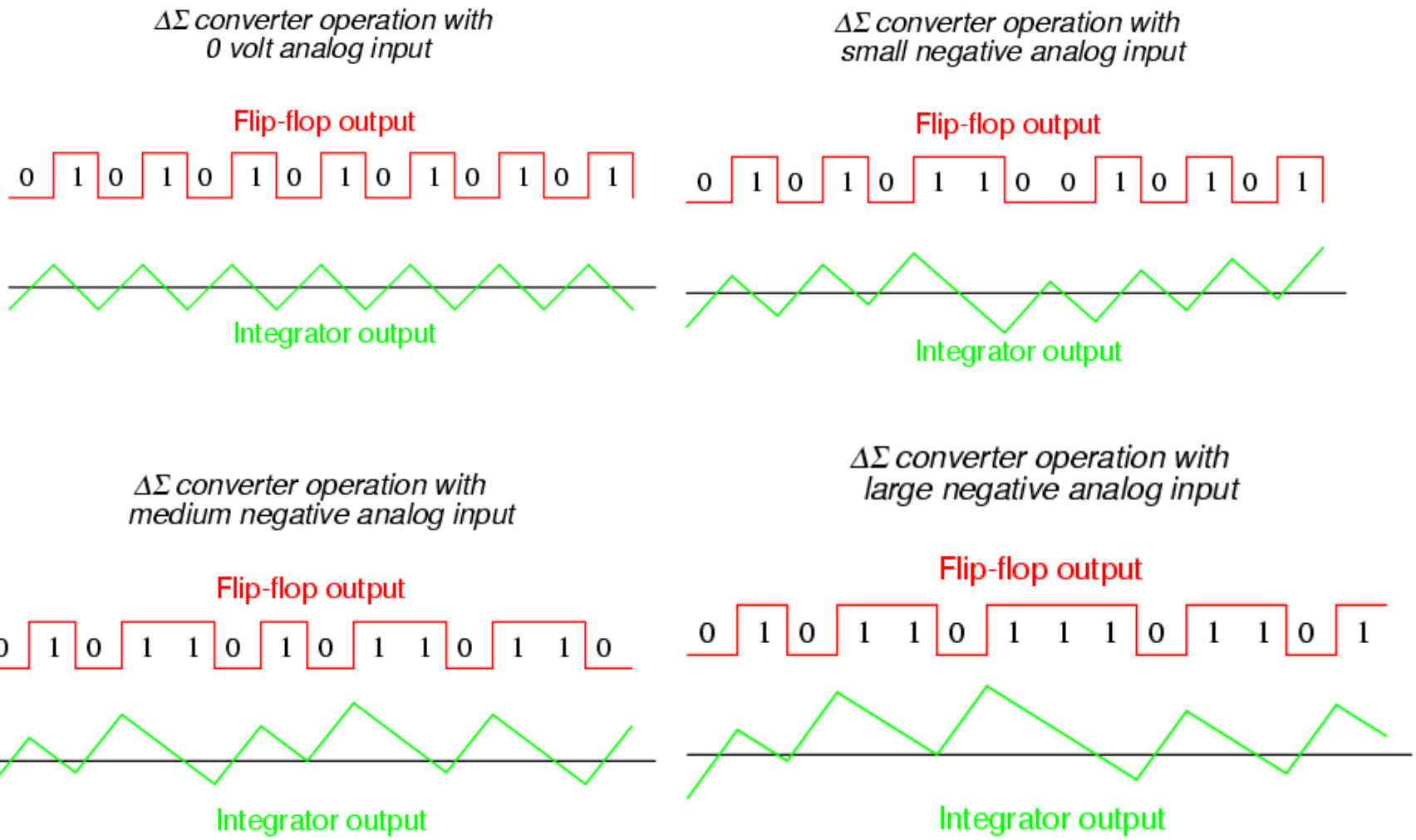
High rate bitstream

Density of 1's at modulator output proportional to the input signal.

Modulator operation

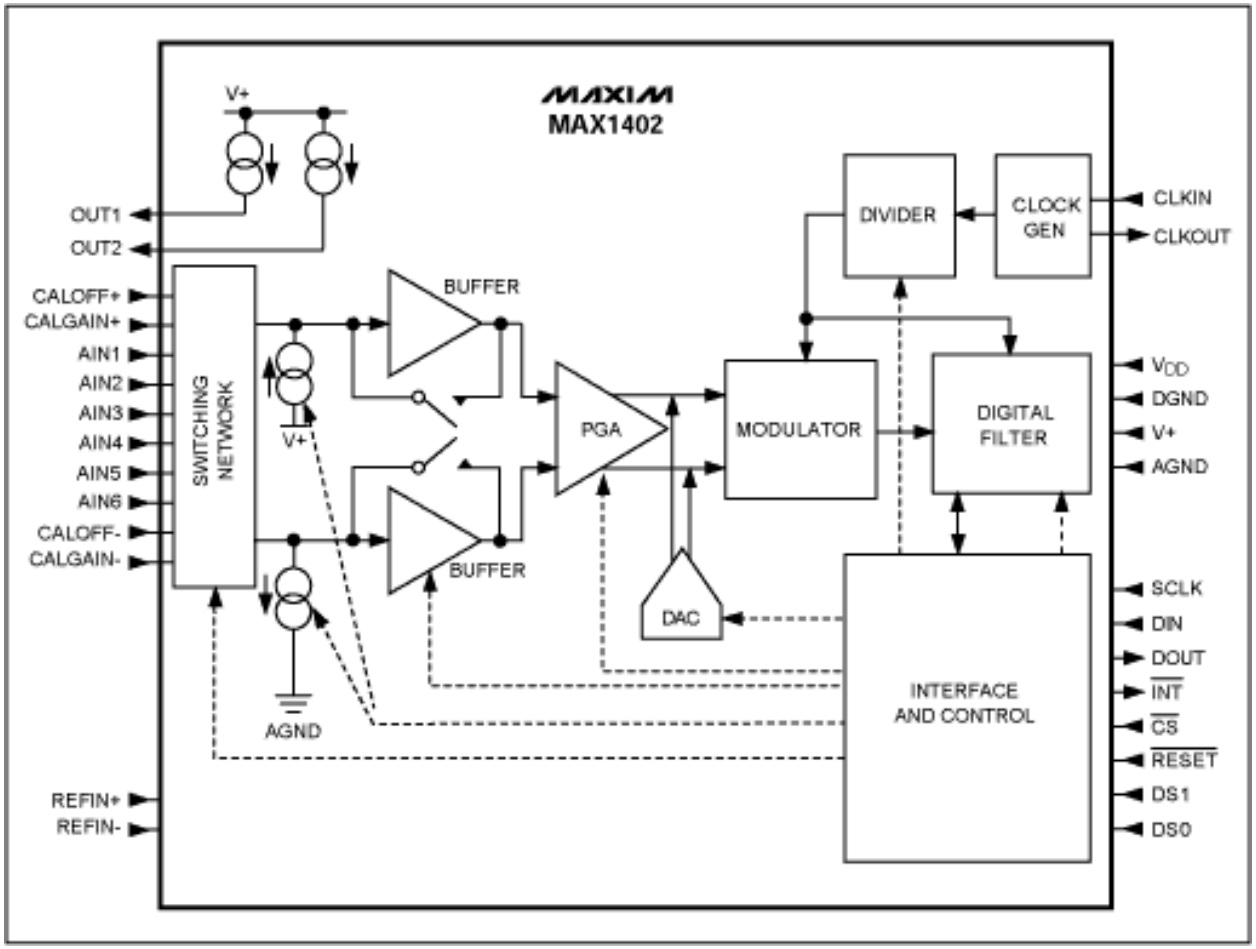
- Slope of integrator output depends on magnitude of Vin
  - “sigma” => summing/integration
- Compare integrator output to 0v, producing “1” if positive and “0” if negative (1-bit ADC)
  - “delta” = difference
- Density of 1’s in the bitstream proportional to magnitude of input voltage Vin

Example



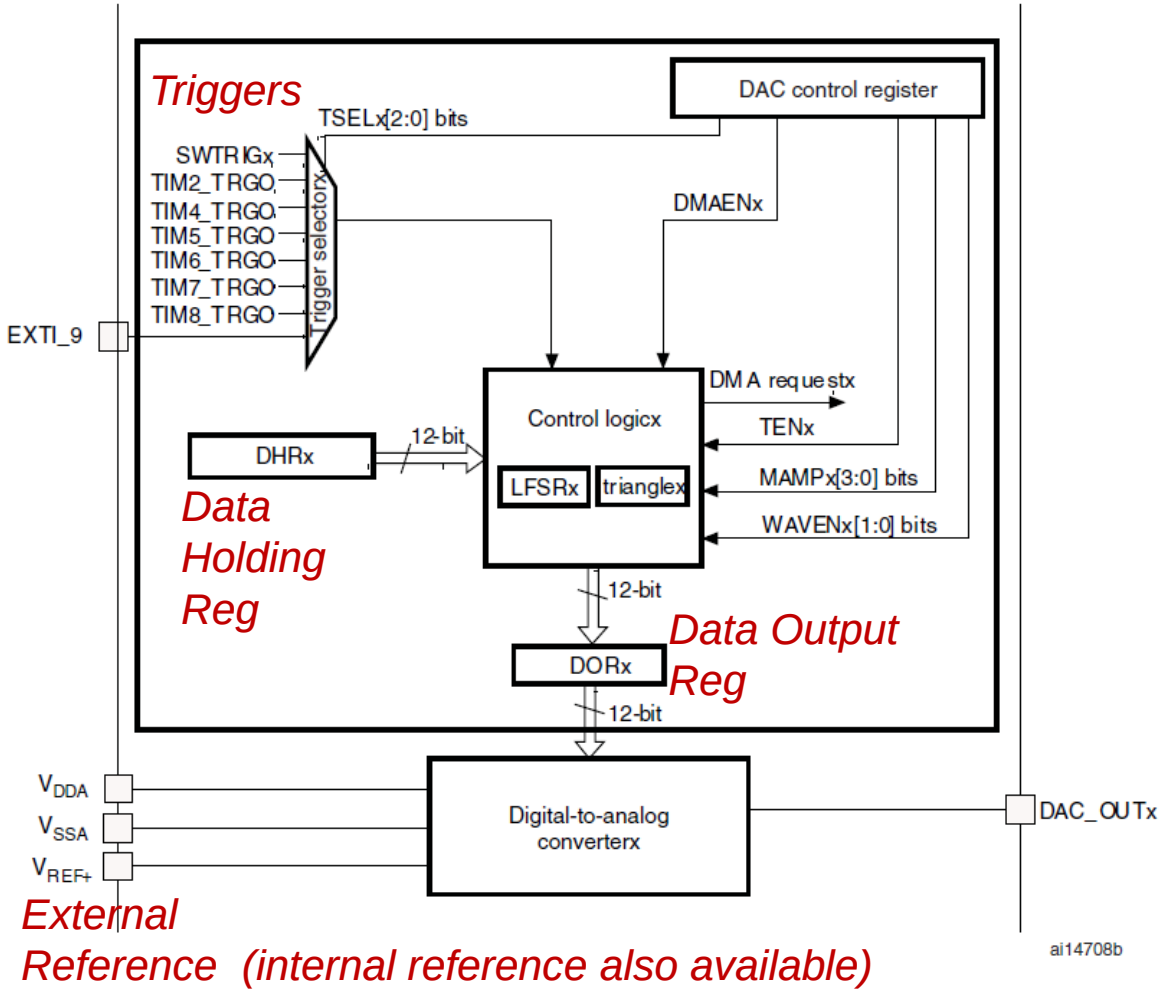
Filtering determines average voltage (density of 1s) in bitstream

Maxim MAX1402 Sigma-Delta ADC



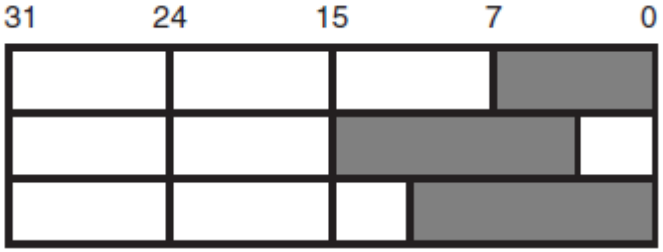
STM32F4xx D/A converter

- 8 or 12-bit modes
- 2 DACs/channels
  - Left/Right channel
  - Concurrent conversions
- Sample triggers:
  - SW trigger
  - Timer triggers
  - EXTI trigger
- DMA support
  - Memory -> DHRx



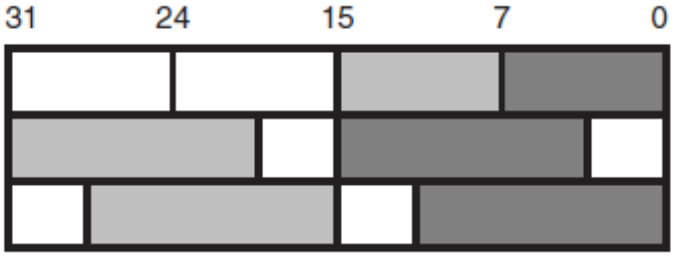
DAC data formats

Single DAC Channel



DHR name (x = 1/2 channel):
DAC_DHR8Rx
DAC_DHR12Lx
DAC_DHR12Rx

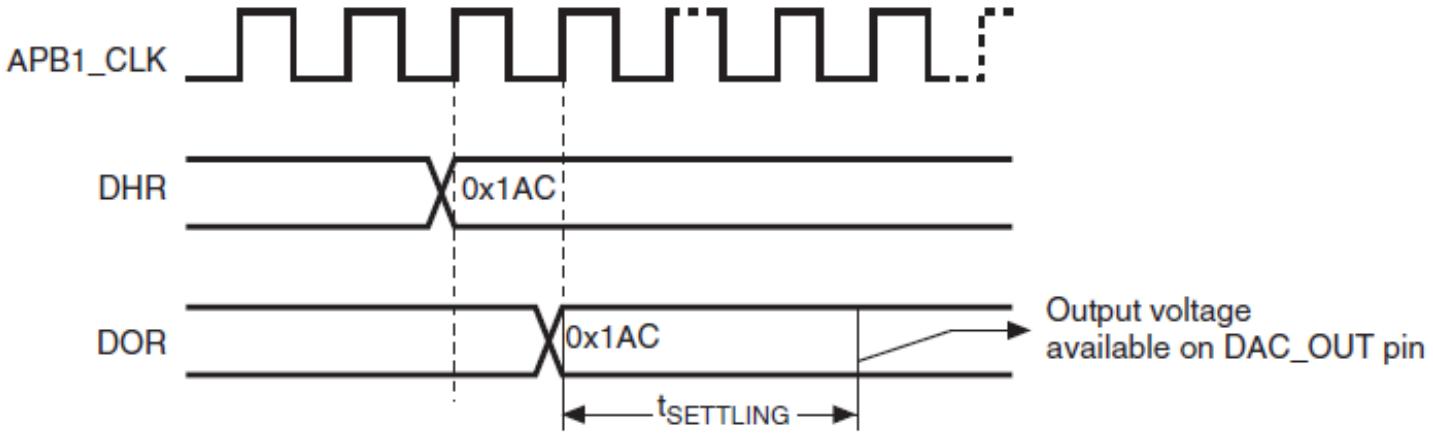
Dual DAC Channels



DHR name:
DAC_DHR8RD
DAC_DHR12LD
DAC_DHR12RD

DAC data conversion

Write data to DAC_DHRx register (trigger disabled: TEN=0):



$$DACoutput = V_{REF} \times \frac{DOR}{4095}$$

DAC control/status registers

DAC_CR (Upper half = channel 2; Lower half = channel 1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2[1:0]		TSEL2[2:0]			TEN2	BOFF2	EN2
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DMAUDRIE1	DMAEN1	MAMP1[3:0]				WAVE1[1:0]		TSEL1[2:0]			TEN1	BOFF1	EN1
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Trigger Select

Trigger Enable*

Channel Enable

* If TEN=0, start when DHR written

DAC_SWTRIGR = Software trigger – start when bit set by SW (reset by HW)

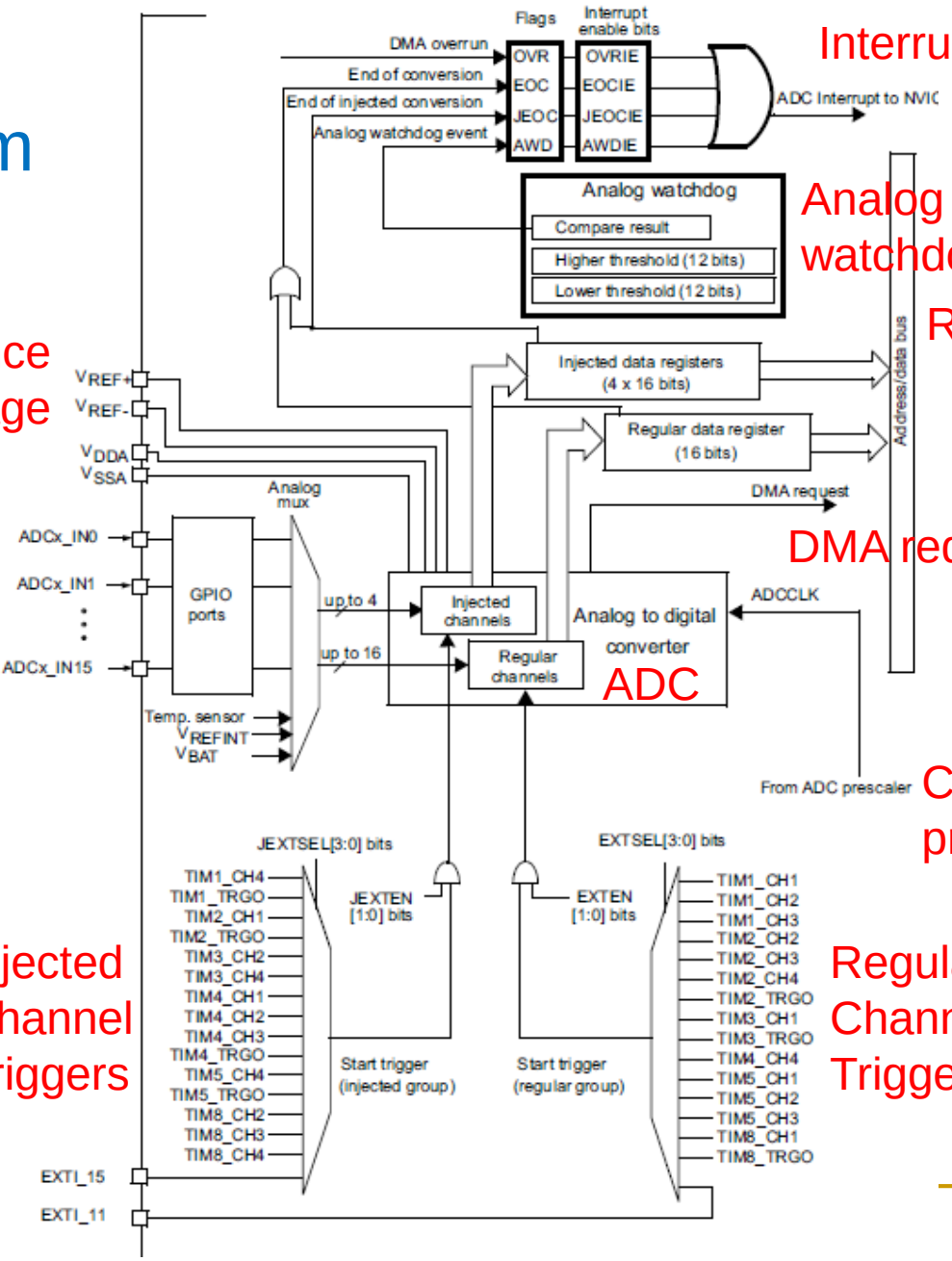
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														SWTRIG2	SWTRIG1
														w	w

DAC_SR = Status Register – Indicates DMS underrun (no data before trigger)

STM32F4xx Successive-Approximation ADC

- 12-bit successive approximation A/D converter
  - Programmable precision: 6-8-10-12 bits
  - Conversion time = #bits + 3 clock cycles
    - 1.2 Msamp/sec @V_{DDA}=1.8-2.4v
    - 1.4 Msamp/sec @V_{DDA}=2.4-3.6v
- “Regular” and “Injected” channel groups
  - Injected channels processed after, or between, regular channels
- 19 multiplexed input channels
  - 16 external sources
  - 3 internal sources: V_{BAT},V_{REFINT}, temperature sensor
- External trigger option (16 sources)
- Multiple conversion modes
  - Single, continuous, scan, discontinuous
- DMA and/or interrupts are supported
  - DMA often used in “scan” mode, to unload the single data register

STM32 ADC block diagram



ADC clocking

- Analog circuitry clock: ADCCLK
  - Derive from APB2 clock ÷ prescale
    - $f_{ADC} = f_{PCLK2}/2$, /4, /6, /8 (bits ADCPRE in ADC_CCR)
    - f_{ADC} required range = 0.6MHz – 18MHz (V_{DDA} = 1.8 to 2.4v)
      = 0.6MHz – 36MHz (V_{DDA} = 2.4 to 3.6v)
  - Sample time (ts) = 3 to 480 clock cycles (8 choices)
    - ts = 0.10µs to 16µs @f_{ADC}=30MHz
    - Set for *each channel* in ADC_SMPR1, ADC_SMPR2
  - Conversion time = ts + n (#data bits) = 9 to 492 clocks
    - 0.50µs to 16.40µs for 12-bit data @f_{ADC}=30MHz
    - fs ≤ 2 Msamples/sec @f_{ADC}=30MHz, ts = 3 ADC cycles
- Enable HSI clock in RCC->CR, which runs ADC conversions
  - RCC->CR |= RCC_CR_HSION; //HSION = bit 0 of RCC->CR
- Digital interface clock (register read/write)
  - Enable APB2 clock in RCC_APB2ENR (clock enable register)

Conversion modes

- **Single conversion** (default: SCAN=0 in CR1, CONT=0 in CR2)
  - Select an input channel (SQ1 field in in **ADC1->SQR5**)
  - Start the conversion (software start or hardware trigger)
  - EOC sets when conversion is complete
  - Read the result in the DR
- **Scan mode** (enable with SCAN=1 in CR1)
  - Perform a **sequence** of conversions of designated input channels
    - Define sequence length in **ADC1->SQR1**
    - Select channels in **ADC1->SQR1**…**ADC1->SQR5** (channels can be in any order)
  - Start the conversion sequence (software start or hardware trigger)
  - EOC sets after each conversion (EOCS = 0) or after the entire sequence is complete (EOCS = 1). EOCS is in **ADC1->CR2**
- **Continuous mode** (enable with CONT=1 in CR2)
  - Start 1st conversion/sequence (software start or hardware trigger)
  - Next conversion/sequence starts automatically after a conversion/sequence completes

Scan mode

- Convert multiple channels in a “sequence”
  - Enable via SCAN bit in ADC_CR1
  - Repeat if CONT bit set in ADC_CR2
  - EOC bit set in ADC_SR at end of sequence or after each conversion (select via EOCS bit)
    - Regular channel data to ADC_DR
    - Injected channel data to ADC_JDR1 – ADC_JDR4
  - Configure sequence via sequence registers
    - ADC_SQR1 – seq. length and channel #s for conversions 13-16
    - ADC_SQR2 – channel #s for conversions 7-12
    - ADC_SQR3 – channel #s for conversions 1-6
    - ADC_JSQR – seq. length and channel #s for up to 4 injected channels
  - If JAUTO=1 (in ADC_CR1),
    - Injected group is converted after regular group after regular trigger
    - Injected group interrupts regular group after injection trigger

Discontinuous mode

- Convert a subset of a sequence on each external trigger
- Regular group, on external trigger:
  - convert n (≤ 8) channels from a sequence
  - convert the next n channels on the next trigger
  - repeat until all channels in the sequence are done
  - restart the sequence on the next trigger
- Injected group:
  - Similar, but only 1 channel per external trigger

STM32 ADC control register 1(ADC_CR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					OVRIE	RES		AWDEN	JAWDEN	Reserved					
					rw	rw	rw	rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			JDISCEN	DISCEN	JAUTO	AWDSGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

RES: resolution (00=12 bit, 01=10-bit, 10=8-bit, 11=6-bit)
SCAN: enable scan mode (channel #s in ADC_SQRx, ADC_JSQRx)
JAUTO: enable automatic injected group conversion after regular group
Interrupt enables:
EOCIE/JEOCIE: on end of conversion (regular/injected channel)
OVRIE: on overrun
Discontinuous mode:
DISCEN/JDISCEN: enable on regular/injected channels
DISCNUM: # channels to convert after trigger (1-8)
Analog Watchdog:
AWDEN/JAWDEN: enable on regular/injected channels
AWDCH: analog watchdog channel selection
AWDSGL: enable watchdog on single channel in scan mode
AWDIE: enable interrupt on analog watchdog

STM32 ADC control register 2(ADC_CR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reserved	SWSTART	EXTEN		EXTSEL[3:0]				reserved	JSWSTART	JEXTEN		JEXTSEL[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved				ALIGN	EOCS	DDS	DMA	Reserved						CONT	ADON
				rw	rw	rw	rw							rw	rw

ADON: 1=enable ADC, 0=disable ADC and power down

CONT: 1 = continuous conversions, 0 = single conversion

ALIGN: data alignment in 16-bit data register (0=right, 1=left)

EOCS: end of conversion selection

0=set EOC at end of sequence, 1=set EOC at end of each conversion

DMA: DMA enable

DDS: DMA disable selection

0=no new DMS request after last channel,

1=continue DMA requests as long as DMA=1

SWSTART/JSWSTART: start conversion of regular/injected channels

EXTEN/JEXTEN: external trigger event

00=disable, 01=rising edge, 10=falling edge, 11=both edges

EXTSEL/JEXTSEL[3:0]: select external event for trigger (regular/injected)

different sets of 16 sources for regular and injected mode

ADC status register ADC_SR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										OVR	STRT	JSTRT	JEOC	EOC	AWD
										rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

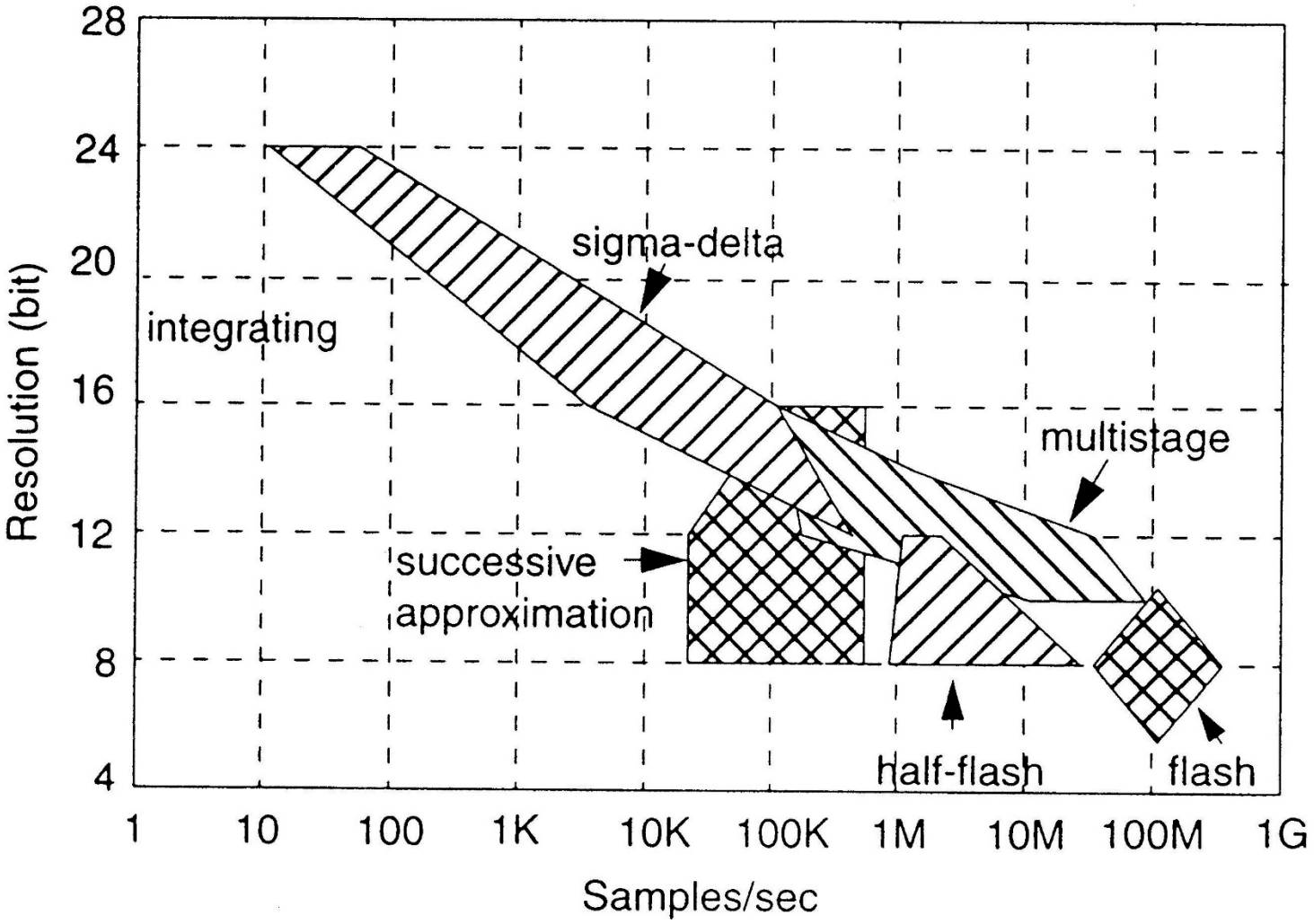
OVR: overrun flag (set if data has been lost)
STRT/JSTRT: regular/injected channel conversion started flag
EOC/JEOC: end of conversion flag (regular/injected channel)
End of sequence (if EOCS=1) or one conversion (EOCS=0)
AWD: analog watchdog flag
“event: if voltage crosses values in ADC_LTR and ADC_HTR

All flags set by HW and cleared by SW

ADC converter characteristics

Type	Need SHA ?	Cycles/ conversion	Advant-ages	Disadvant-ages	Example
Flash	No	1	Fastest	Expensive, power	6-bit @ 400MHz
Successive Approx-imation	Yes	>= 2	Fast, cheap	Slower than flash	8-bit @ 20 MHz
Integrating	Yes	Varies	Precise	Slow	22-bit @ 20Hz
Sigma-Delta	No	Many	Mostly digital, linear, high resolution	Complex digital circuit	16-bit @ 100 KHz

ADC converter comparison



ADC selection (Analog Devices, Inc.)

- http://www.analog.com/en/analog-to-digital-converters/products/index.html