Practical Digital Signal Processing

Interfacing DSPs and microprocessors to the Real World

W.G.Marshall

Recommended Book

A Simple Approach to Digital Signal Processing

C Marven and G Ewers

Wiley

Course Structure

- Introduction: Analogue & Digital Worlds
- Analogue Signal Conditioning
- Digital Signal Processing
  - DSP Hardware Architecture Issues
  - The TMS 320C50 DSP chip
  - DSP Programming Issues
- Transducers
  - Selection and Interfacing
- Practical Work

Practical Work

- Interface a transducer to DSP system
  - Analogue amplifier design for:
    - Impedance Matching/Buffering
    - Signal amplification
    - Noise removal
    - Anti-alias filtering
- Digital filter selection and programming
- Analogue output to CRO/spectrum analyser to check result.
- Use TFA to check overall phase linearity.

Measuring the World 1

- The **Real** world is Analogue!
  - All practical signals vary continuously both in amplitude and time
  - Examples:
  - When a filament lamp is turned on, the current takes time to rise - it doesn't happen instantaneously.
  - A human heart-beat waveform has 'shape' as well as a repetition frequency.

Measuring the World 2

- Component Parts
  - Transducer(s)
    - Measured parameter converted to electrical signal
  - Signal Processing System
    - Analogue or Digital
  - Output Devices
    - Visual/Audio Interface to humans
    - Feedback for ‘Closed-loop’ Control

Traditional Methods 1

- Single-Unit Measure-Display Systems
  - Little or No processing
    - Examples:
    - Moving-coil current meter
    - Mercury thermometer
    - Mercury barometer
  - All are ‘Analogue’ Systems, converting a parameter directly to a visible display.

Traditional Methods 2

- Two-Part Measure-Display Systems
  - Little or No processing
    - Examples:
    - Car coolant temperature sender and gauge
    - Car fuel level sender and gauge
  - Both are ‘Analogue’ systems, converting a parameter to an electrical signal (current).
  - Both use hot-wire method to convert signal to a pointer position.

Traditional Methods 3

- Measure-Process-Display Systems
  - Signal ‘Conditioning’ circuits added
    - Example:
    - Hi-fi system using vinyl LPs
    - Magnetic cartridge converts stylus vibration to electrical current.
    - Input circuits of amplifier boost signal and perform frequency equalization
    - Power amplifier drives audio transducers which convert electrical signals to sound (Loudspeaker)

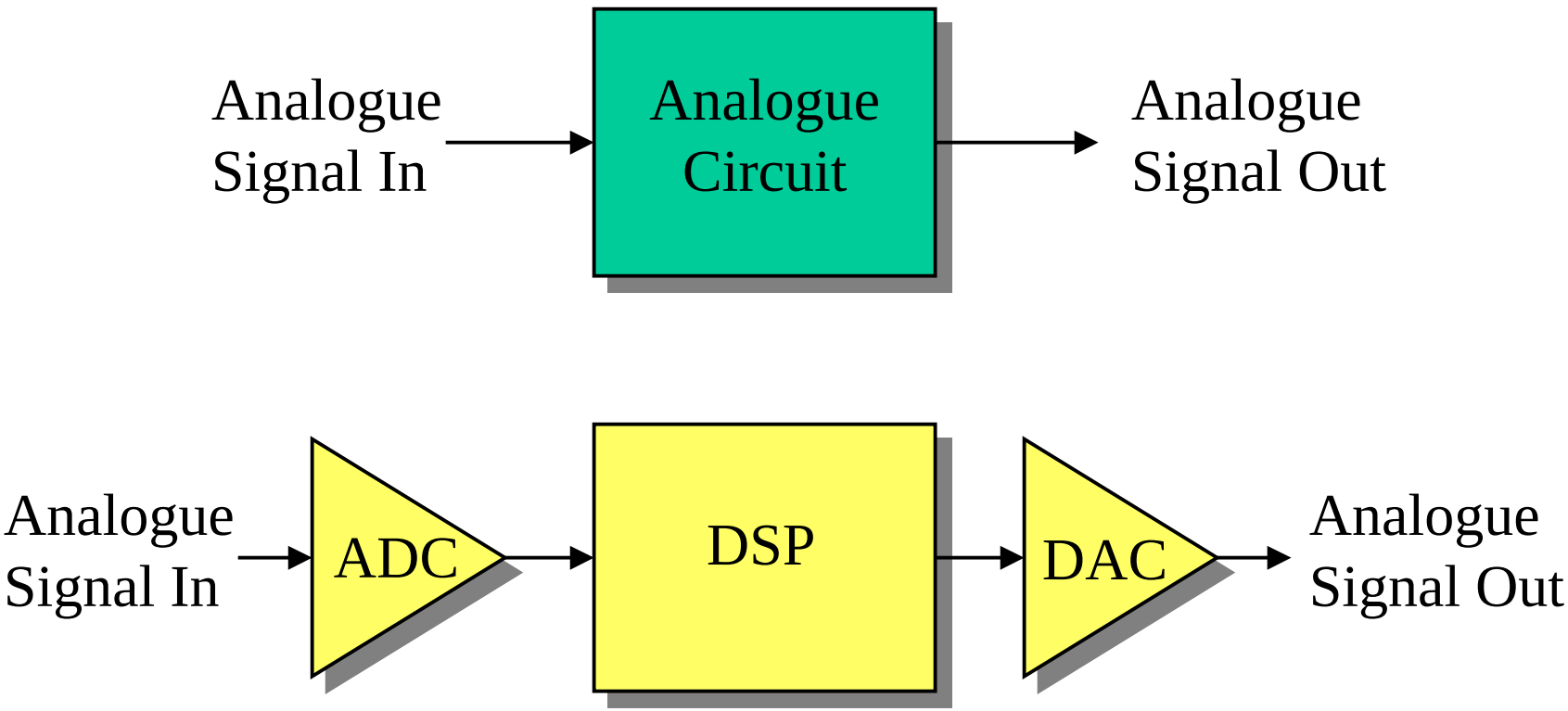
Problems with Tradition

- Component Tolerances
- Component Aging
- Electrical Noise Problems
- Complex hardware design for even simple processing tasks
- Difficult to debug/modify/update processing
- Design compromises inevitable

Advantages of DSP

- Component tolerances largely irrelevant
- Ageing only very long term problem
- Noise problems reduced
- Hardware exchanged for software
- Updates/debugging easier/cheaper
- New digital algorithms not possible using analogue methods

Analogue v Digital Hardware



Signal Conditioning - What for?

- Signal too small (or too large!)
- Interference (High-Frequency noise) present
- Signal Non-Linear
- AC signal has a DC Offset
- Source/Load impedance matching needed
  - Max voltage transfer: $Z_L >> Z_S$ Often required for low-output sensors.
  - Max power transfer: $Z_L = Z_S$ More common in communications circuits.

Conditioning Processes

- Change Format
  - Light → Frequency
- Change Levels
- Reduce noise interference
  - Band-Pass Filter e.g. Notch Filter at 50 Hz
- Linearize, perhaps digitally
- Band-limit (Anti-aliasing) Low-pass filter
- Protection: Over-voltage, Reverse polarity

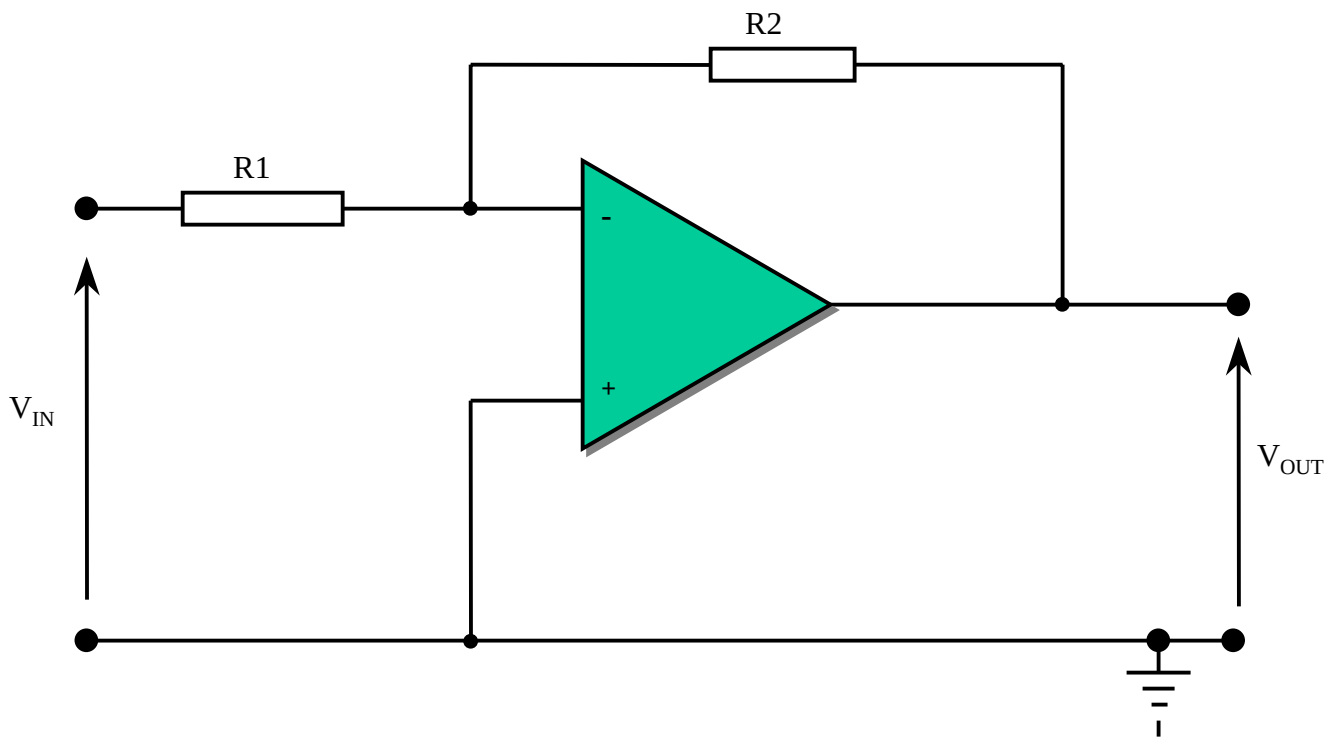
Analogue Conditioning Circuits

- The Operational Amplifier
  - Inverting amplifier
  - Buffer amplifier (voltage follower)
  - Differential amplifier
  - Instrumentation amplifier
  - Low-Pass Filter
  - Logarithmic amplifier
  - Comparator

Inverting Amplifier

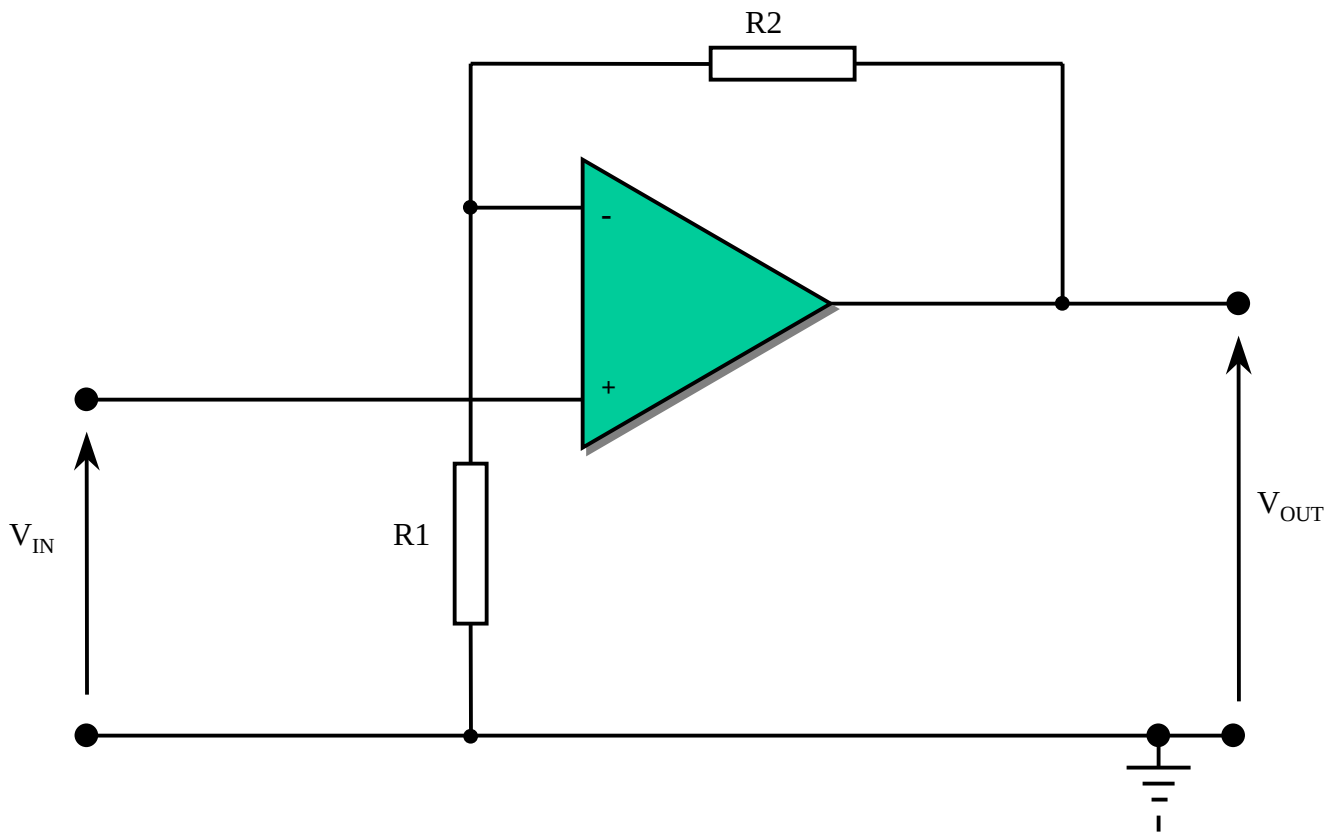
Inverts signal and amplifies

$$\text{Voltage Gain} = \frac{V_{OUT}}{V_{IN}} = -\frac{R_2}{R_1}$$



Non-Inverting Amplifier

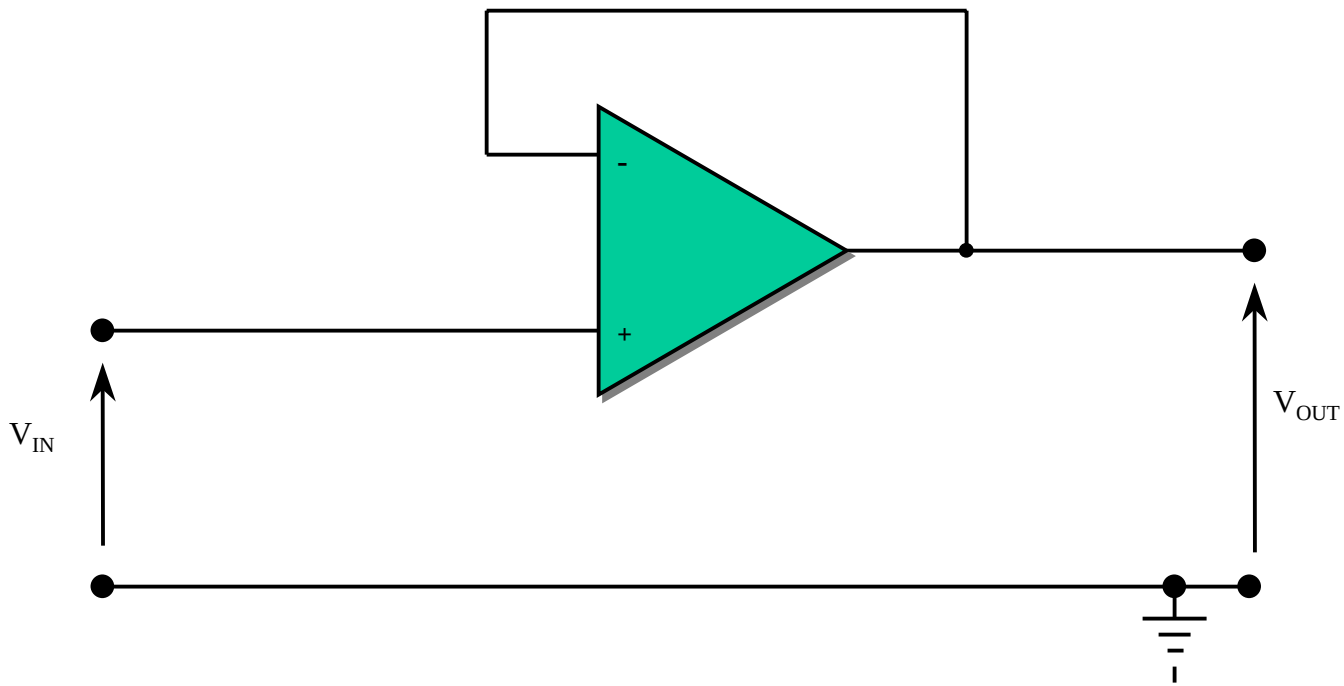
$$\text{Voltage Gain} = \frac{V_{OUT}}{V_{IN}} = 1 + \frac{R_2}{R_1}$$



Buffer Amplifier

Voltage Gain = 1 $V_{OUT} = V_{IN}$

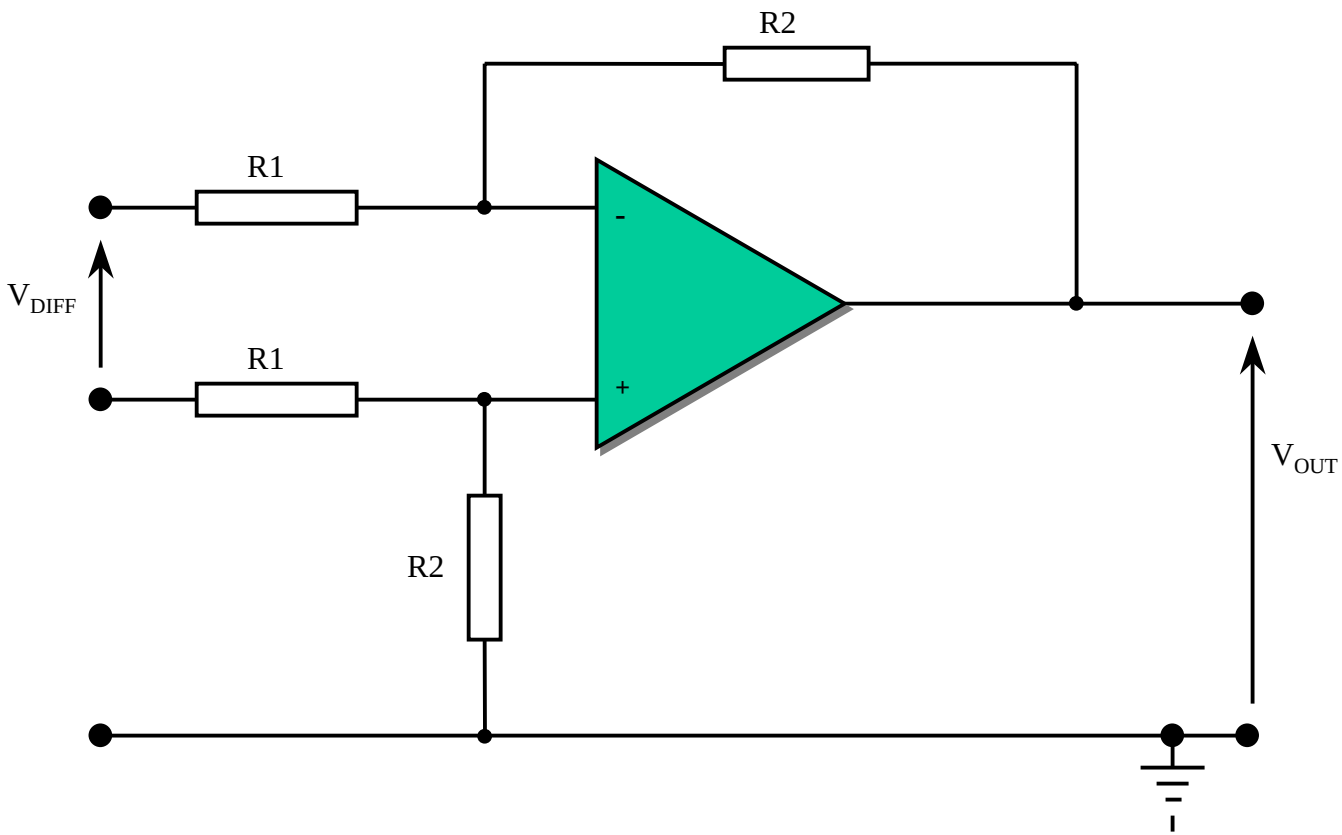
R_{OUT} = Low R_{IN} = High



Differential Amplifier

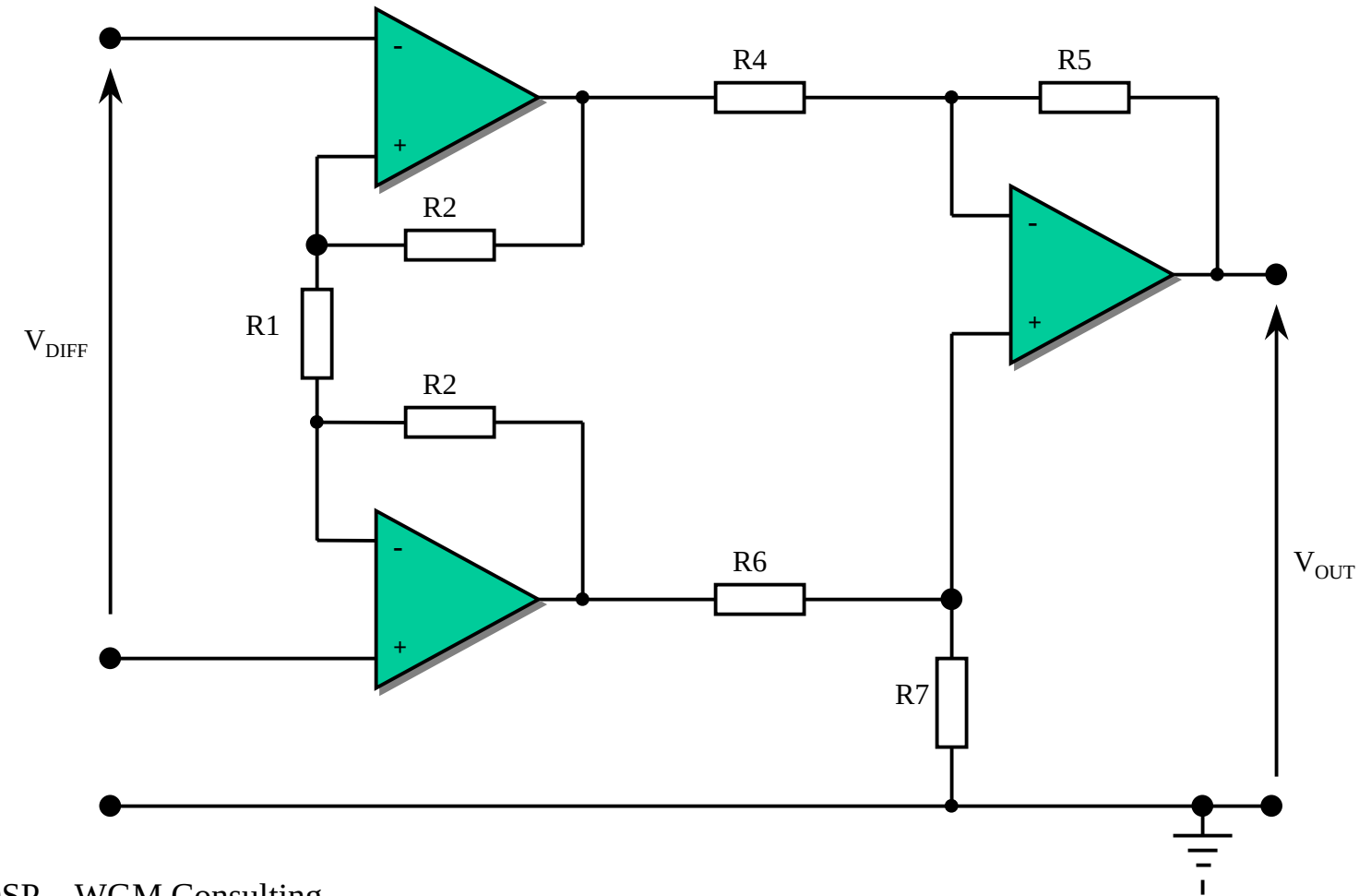
Amplifies the **Difference** between the inputs

$$\text{Voltage Gain} = \frac{V_{OUT}}{V_{DIFF}} = \frac{R_2}{R_1}$$



Instrumentation Amplifier

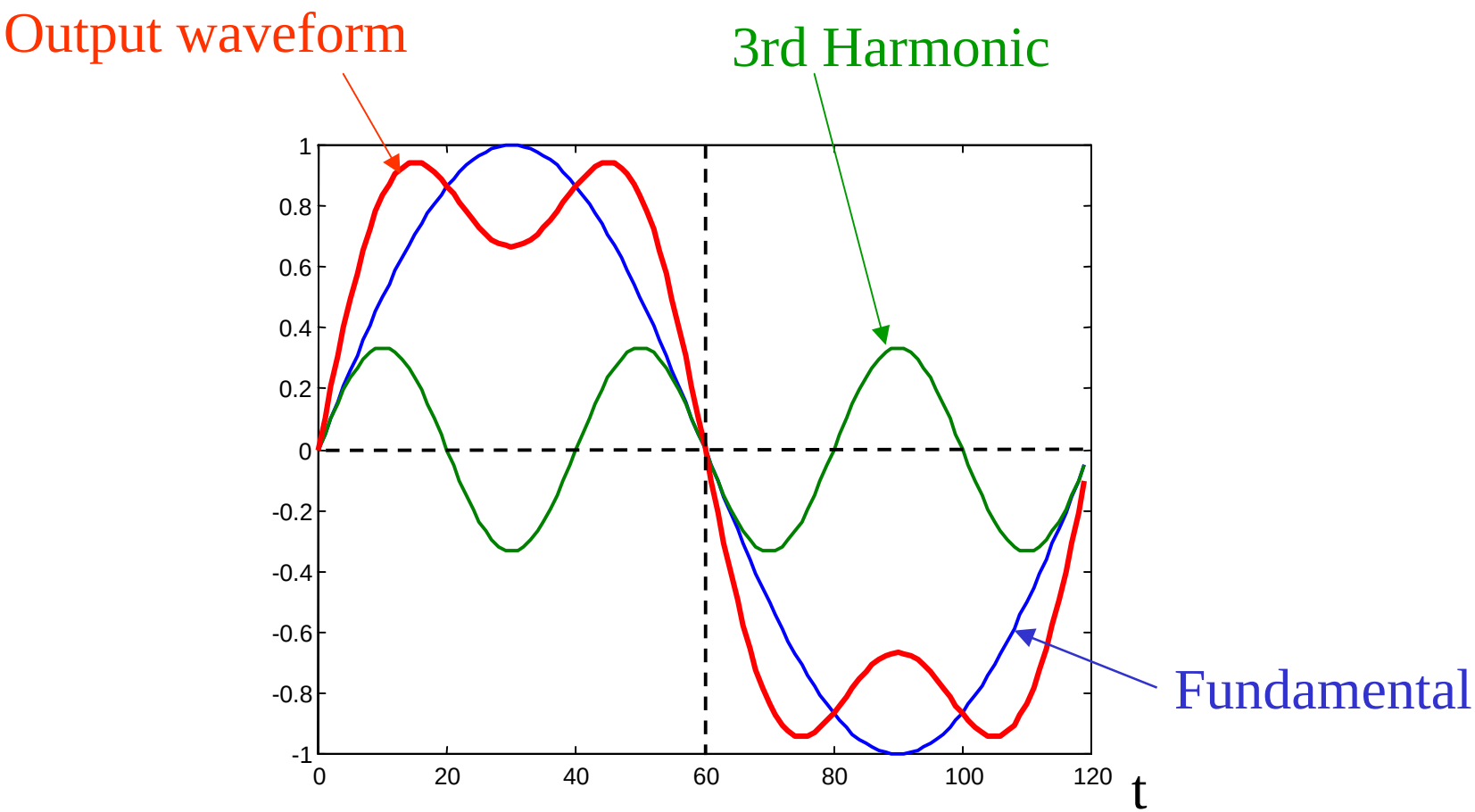
$$\text{Voltage Gain} = \frac{V_{OUT}}{V_{DIFF}} = 1 + \frac{2R_2}{R_1} \qquad \text{where } \frac{R_4}{R_5} = \frac{R_6}{R_7}$$



Filter Design Considerations

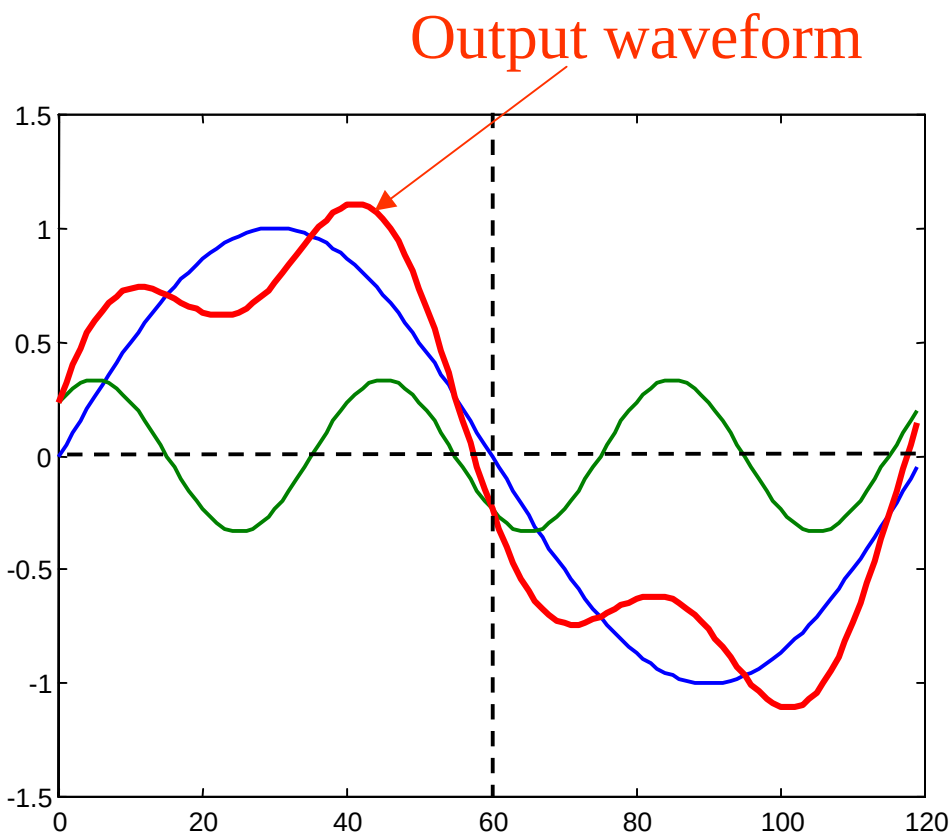
- Flatness of Pass-Band frequency response
  - Flat (Butterworth)
- Slope of ‘Roll-Off’
  - Steep for a given order (Chebyshev)
- Phase Linearity in Pass-Band
  - Near-constant Group Delay (Bessel)
- Hardware Implementation
  - Sallen Key, Multipath Feedback, Twin-Tee, etc

Phase Linearity



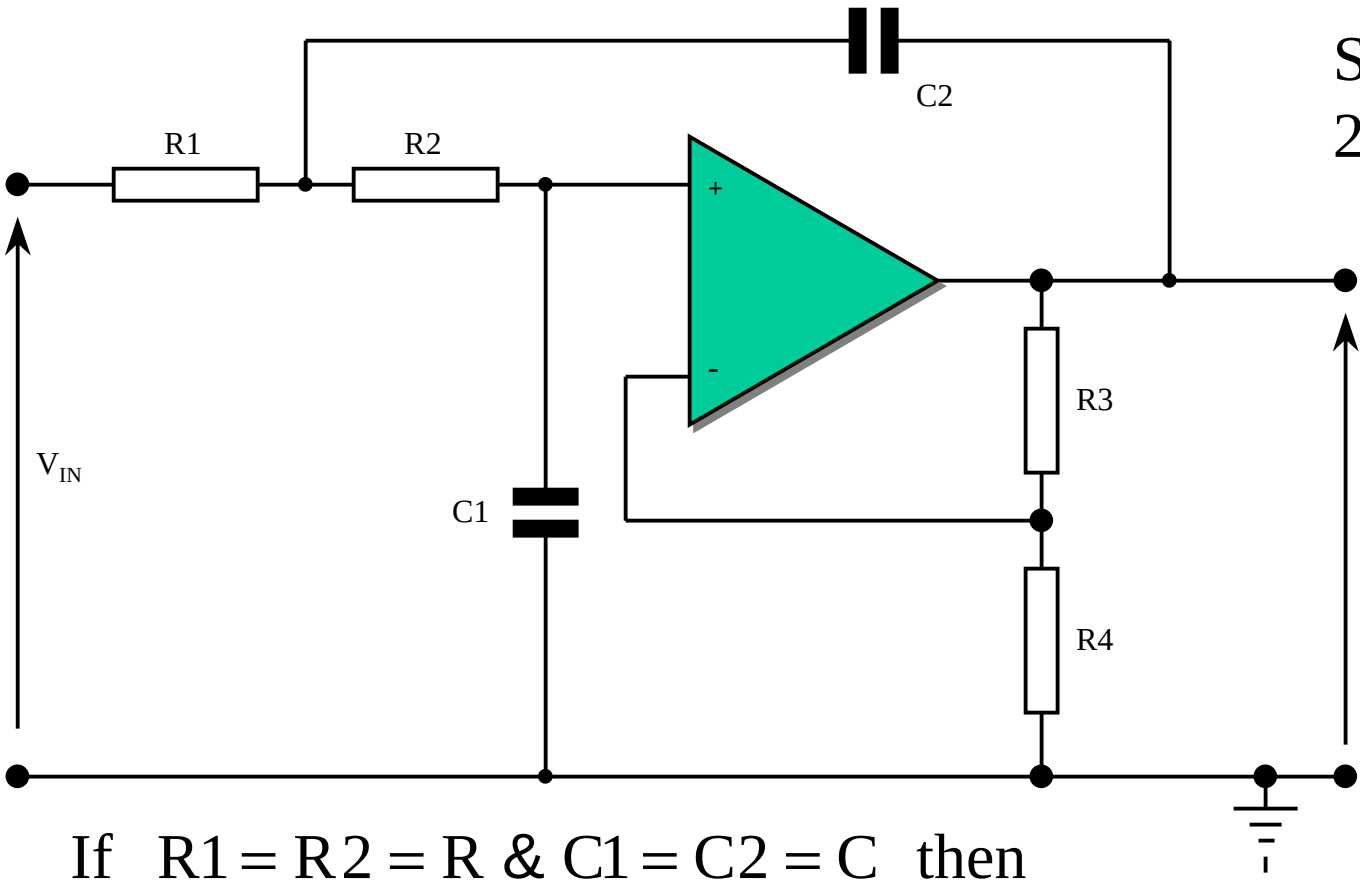
Linear-Phase system: all frequencies travelling at same speed, still in phase at output. Waveform symmetrical.

Effect of Non-Linear Phase



Non-Linear-Phase system: frequencies travelling at different speeds, out of phase at output. Waveform distorted.

Analogue Filter



Sallen-Key
2nd Order LP

If $R1 = R2 = R$ & $C1 = C2 = C$ then

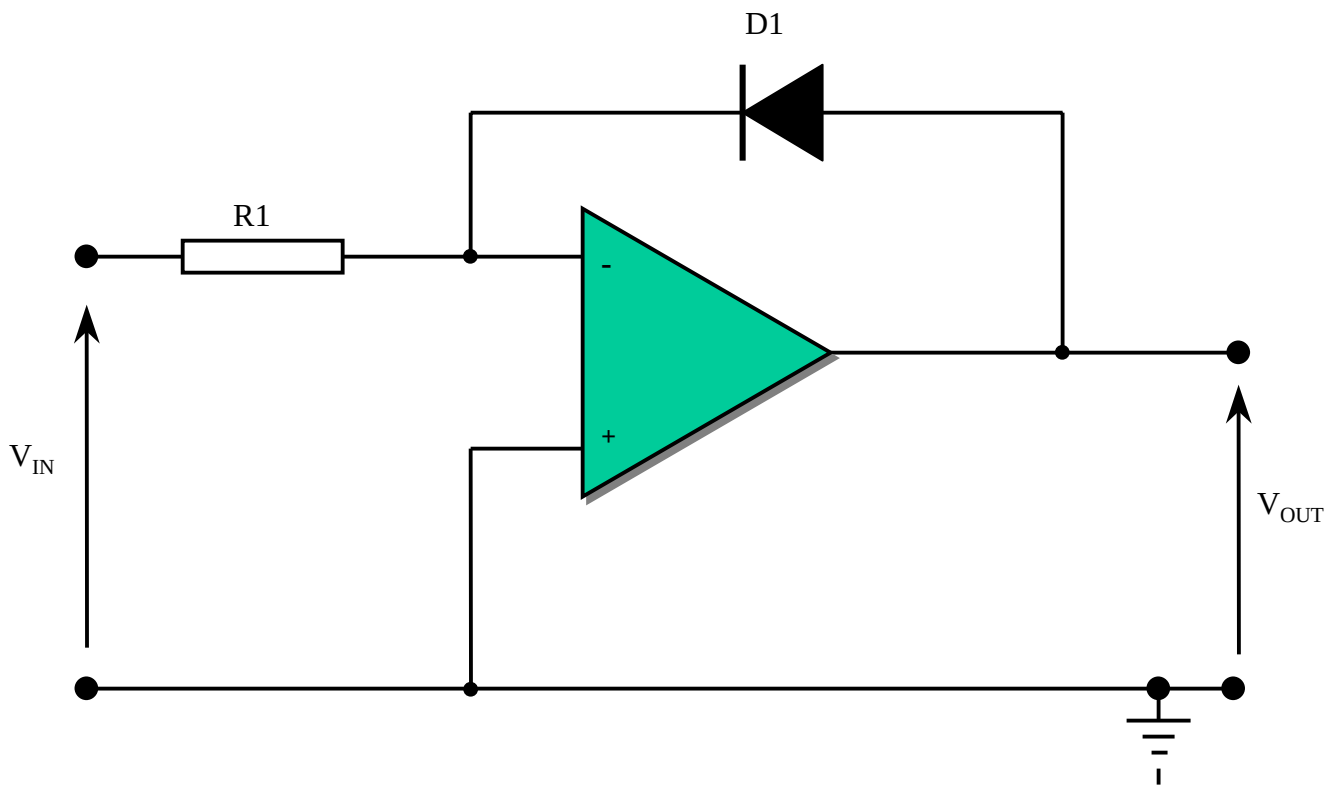
$$f_C = \frac{1}{2\pi RC} \qquad K = 1 + \frac{R3}{R4} \qquad Q = \frac{1}{3 - K}$$

Useful for anti-aliasing filter design.

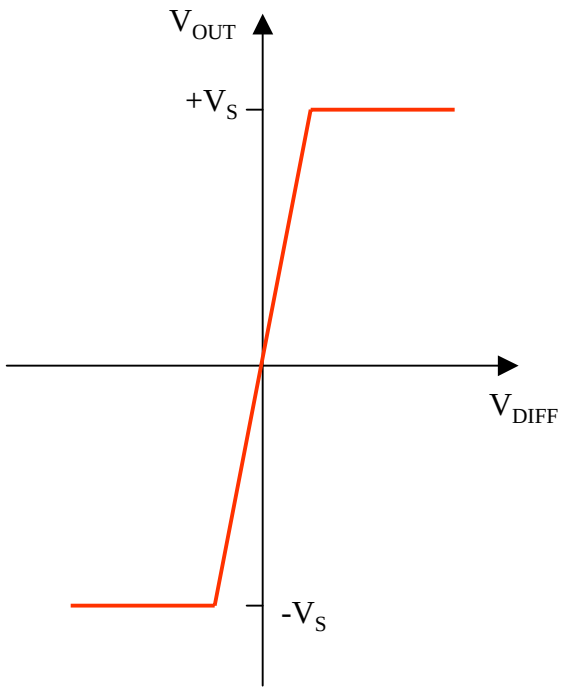
Logarithmic Amplifier

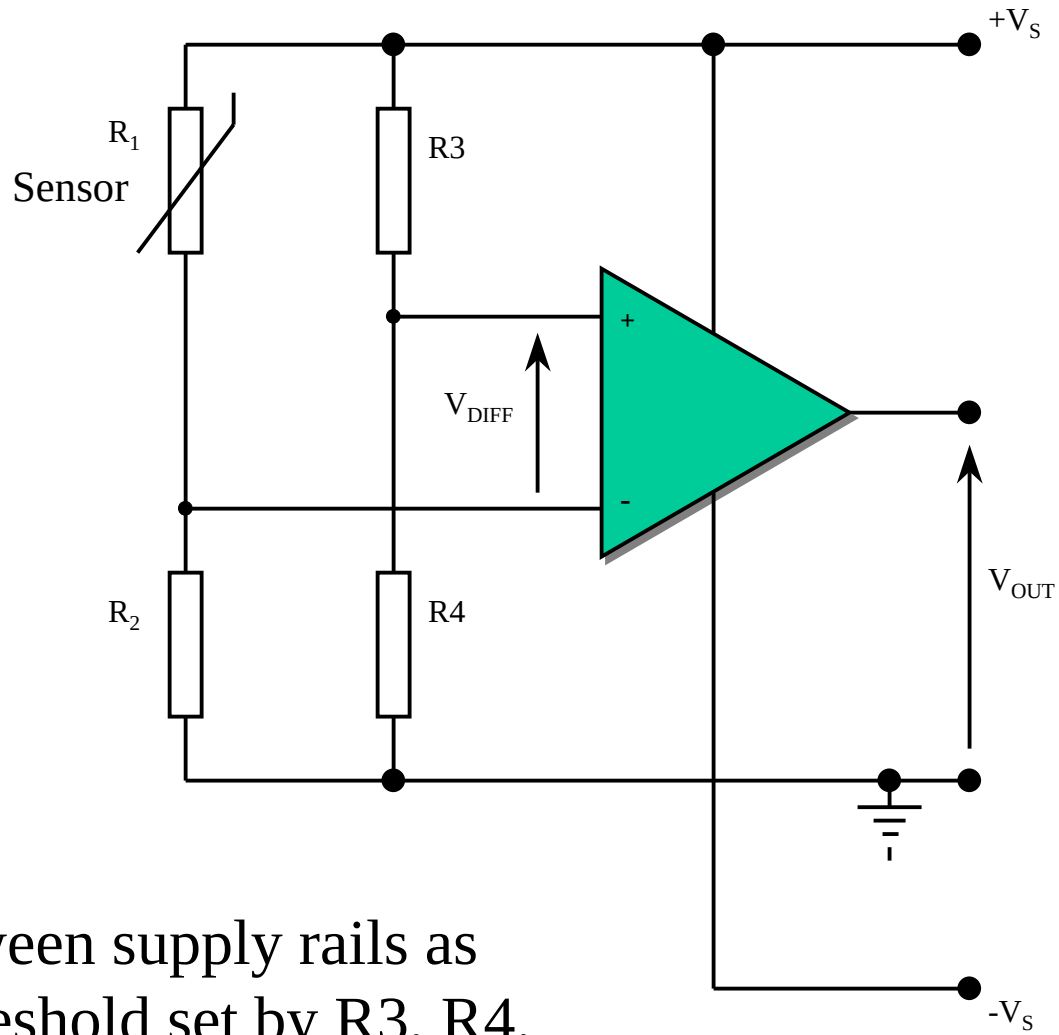
Linearize the output of a non-linear sensor. e.g. Thermocouple

$$V_{OUT} = K \log_e V_{IN}$$



Comparator



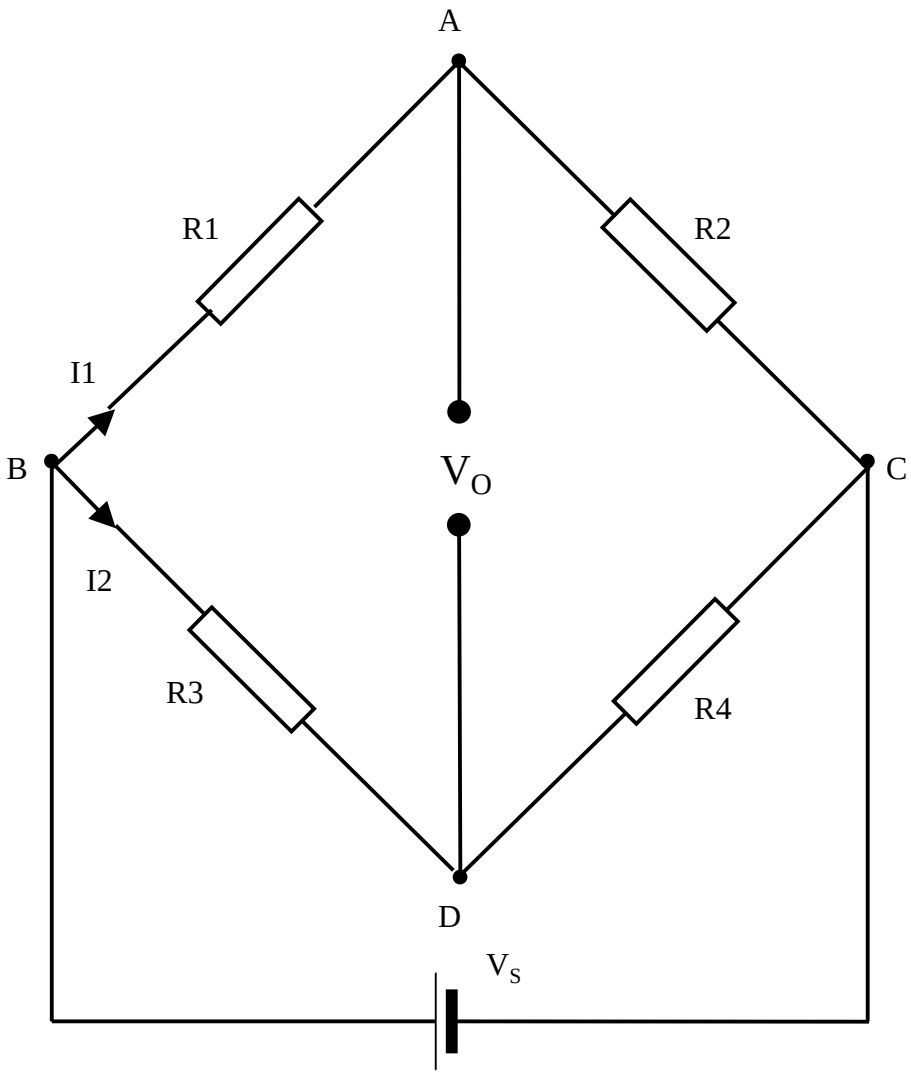


Output switches between supply rails as input moves over threshold set by R3, R4.

Wheatstone Bridge

- Resistive *change* to Voltage *change*
  - Strain gauges
  - Temperature sensors
- Temperature compensation
  - Sensor mounted a distance from bridge.
  - Use 3-wire connection
- High-impedance buffer for V_O required.
  - Differential or Instrumentation amplifier
- Supply V_S must be very stable.

Wheatstone Bridge 2



Bridge is balanced when $V_O = 0$ and

$$\frac{R_1}{R_2} = \frac{R_3}{R_4}$$

$$\Delta V_O \approx V_S \left(\frac{\Delta R_1}{R_1 + R_2} \right)$$

Approximation true, and output linear if: $R_1 >> \Delta R_1$

Wheatstone Bridge Example

R_1 = Resistance type temperature sensor with resistance
R_0 = 150 Ω at 0^O C. All other resistors = 150 Ω.
Temp Coeff for sensor α = 0.0039/K V_S = 6.0 V

Variation of resistance with temperature t : $R_t = R_0 (1 + \alpha t)$

Change in resistance $R_t - R_0 = R_0 \alpha t$

$$= 150 \times 0.0039 \times 1 = 0.585\ \Omega/K$$

Assume V_O measured by high-impedance input device.

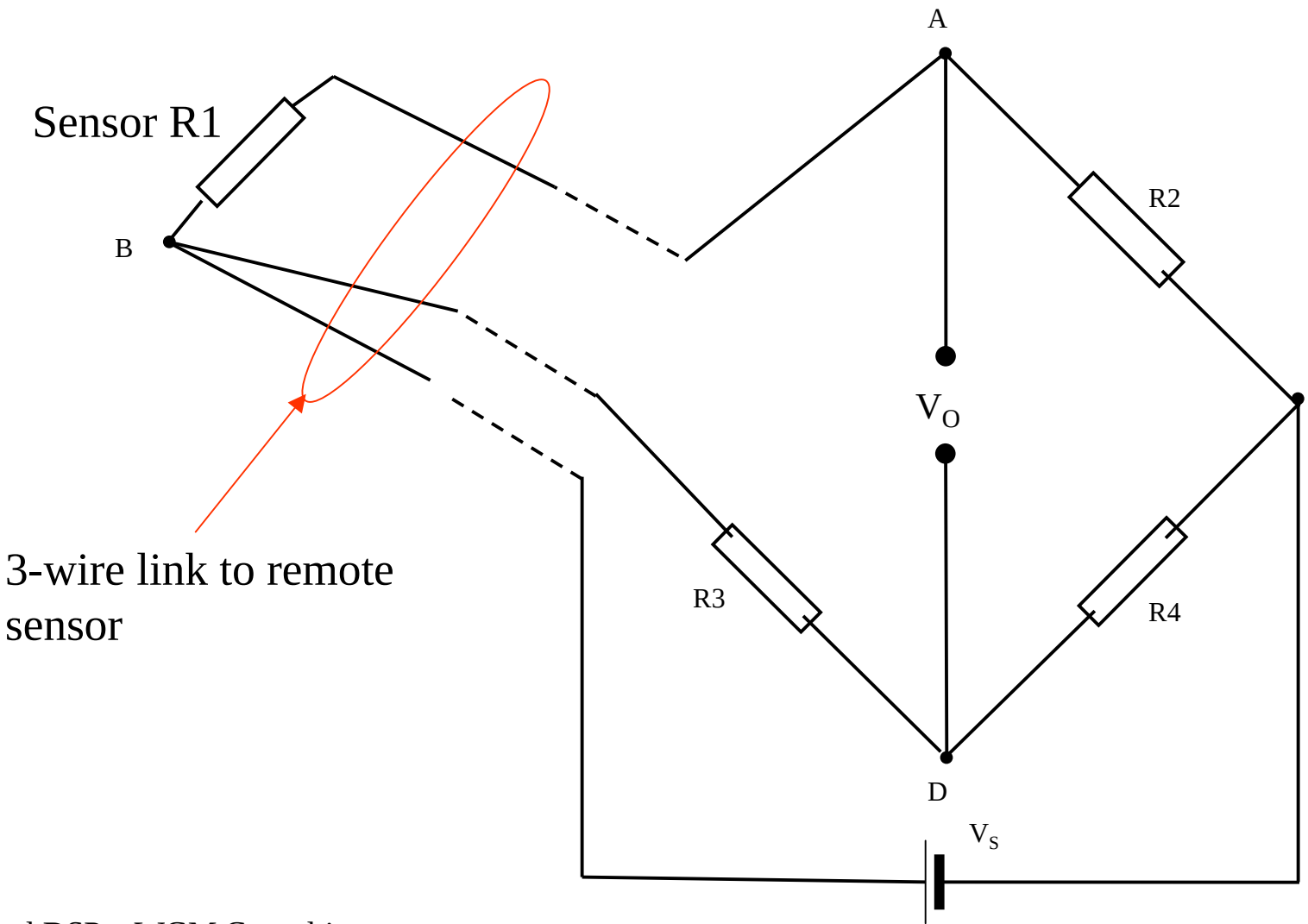
Change is small relative to 150 Ω so use approximation:

$$\Delta V_O \approx V_S \left(\frac{\Delta R_1}{R_1 + R_2} \right) = \frac{6 \times 0.585}{150 + 150} = 0.0117 \ \ V$$

Temperature Compensation

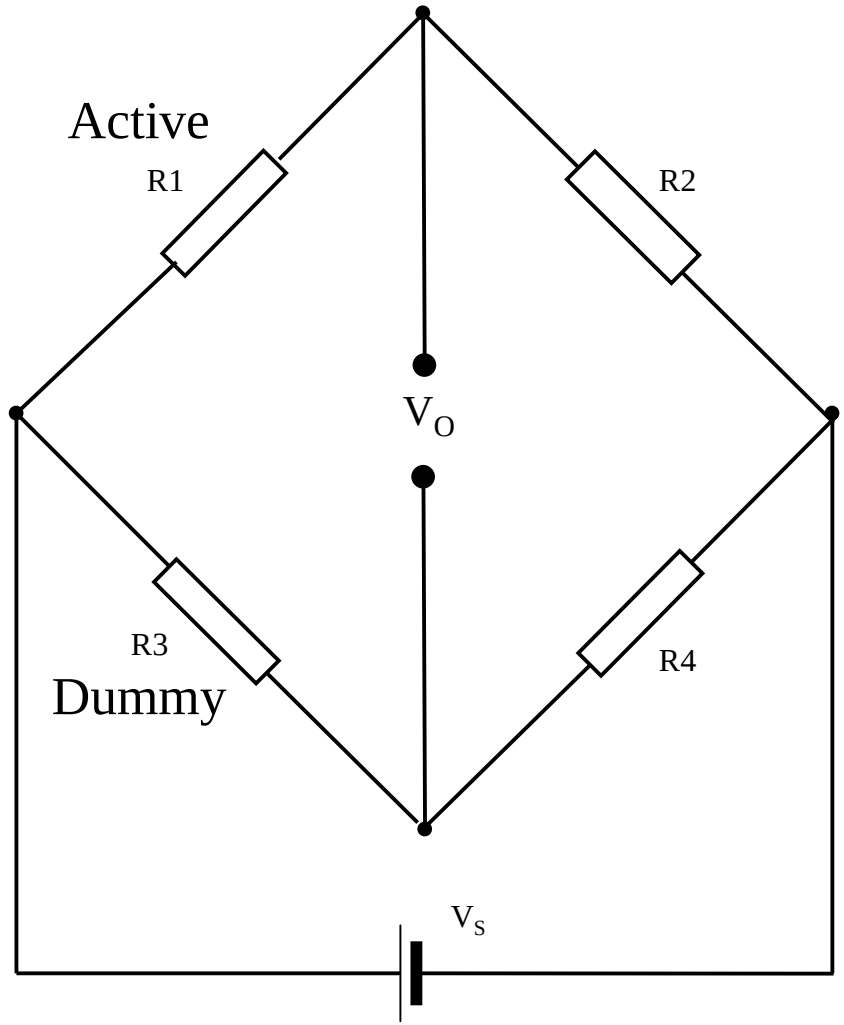
- Sensor often remote from Bridge
- Connecting leads affected by temperature
- Connect sensor with three wires
- Wires are identical and packed together
- One wire in series with Sensor (R1)
- One wire in series with R3
- Third wire is PSU connection

Temperature Compensation 2

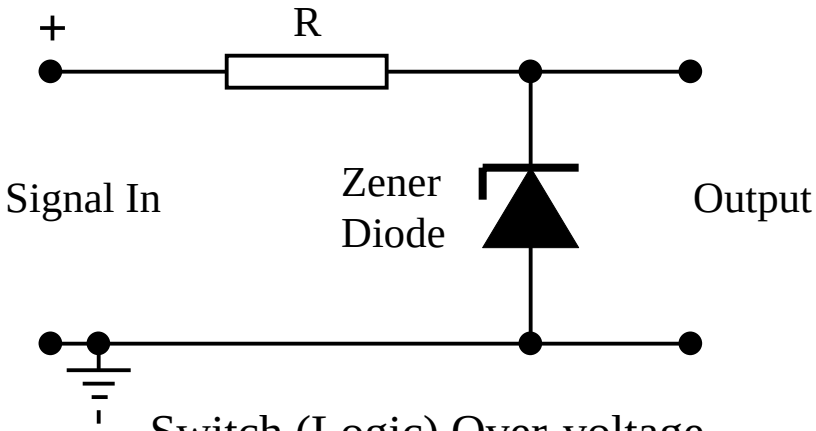


Temperature Compensation 3

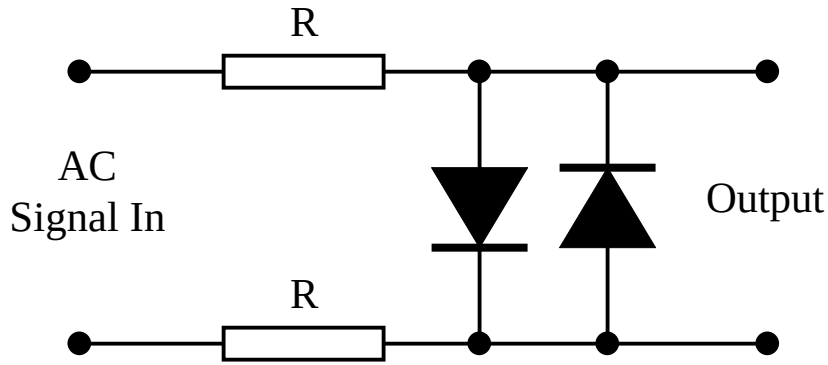
- Strain Gauge Interfacing
  - Resistance change α strain.
  - Resistance also affected by temperature.
- Dummy gauge
  - placed near (but not under strain) to the active gauge.
- Can also use 4 gauges
  - two in compression (R_2, R_3)
  - two in tension (R_1, R_4).
  - Each pair mounted 90^O to the other pair.



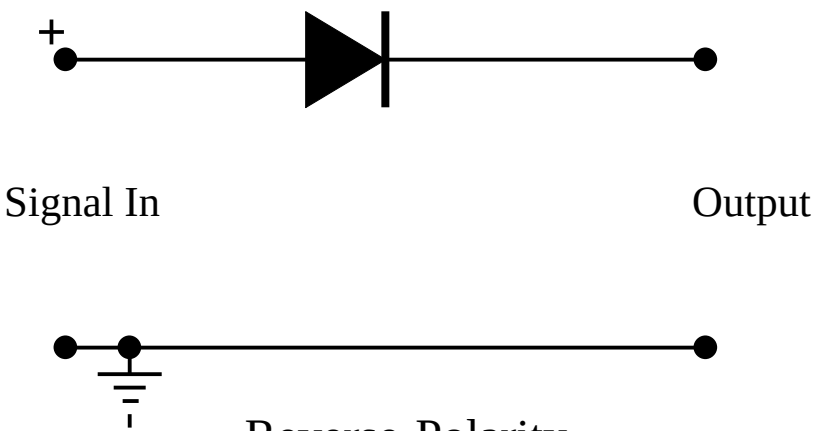
Input Protection



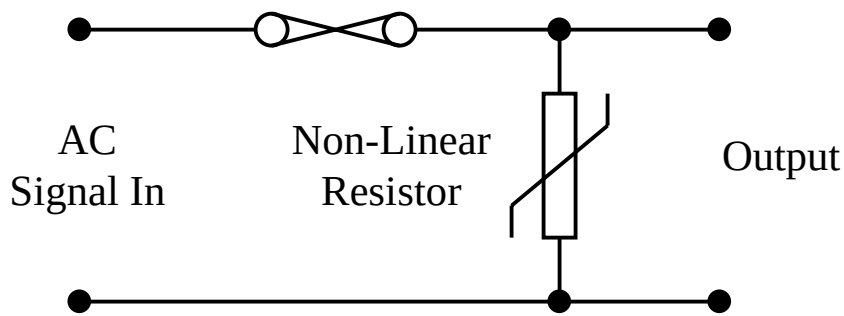
Switch (Logic) Over-voltage



Differential Over-voltage



Reverse-Polarity



Differential Over-voltage

Analogue to Digital

- *Sample* analogue (continuous time) signal at regular (*Sampling*) intervals.
- Use Analogue to Digital Converter (ADC) to convert voltage sample to a number.
- Carry out *processing* on *digitized* samples using a DSP running a *program* version of a mathematical *algorithm*.
- Output samples to a Digital to Analogue Converter (DAC).

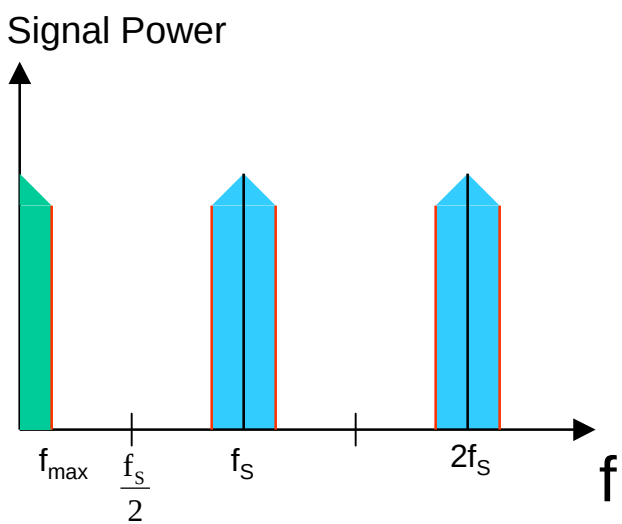
Analogue to Digital Hardware

- Design Considerations
  - Highest frequency components
    - Sampling rate
    - Anti-alias filter
    - Sample and Hold required?
    - Bipolar-Unipolar conv (Input level shift) required?
    - ADC type (Dual-ramp, successive approx or flash)
  - Dynamic Range of signal
    - ADC Resolution (No. of bits)
  - Analogue multiplexing

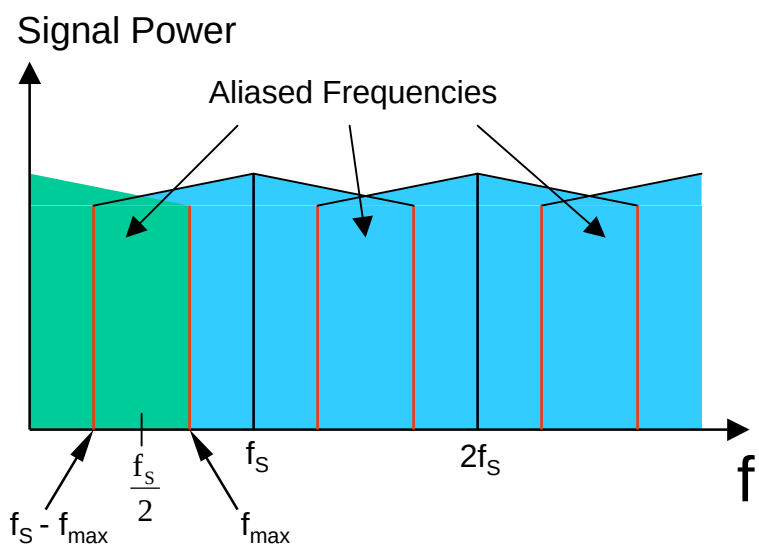
Sampling Rate: Aliasing

- Sampling rate must be at least twice the highest frequency component present in the analogue signal: Nyquist Rate.

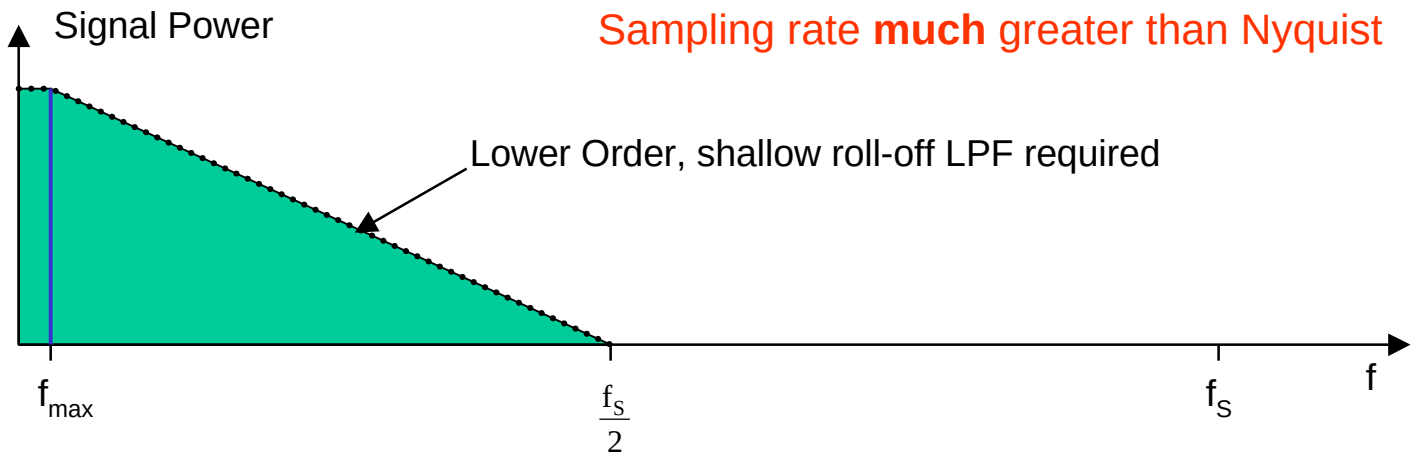
$f_S > 2f_{max}$: No Aliasing

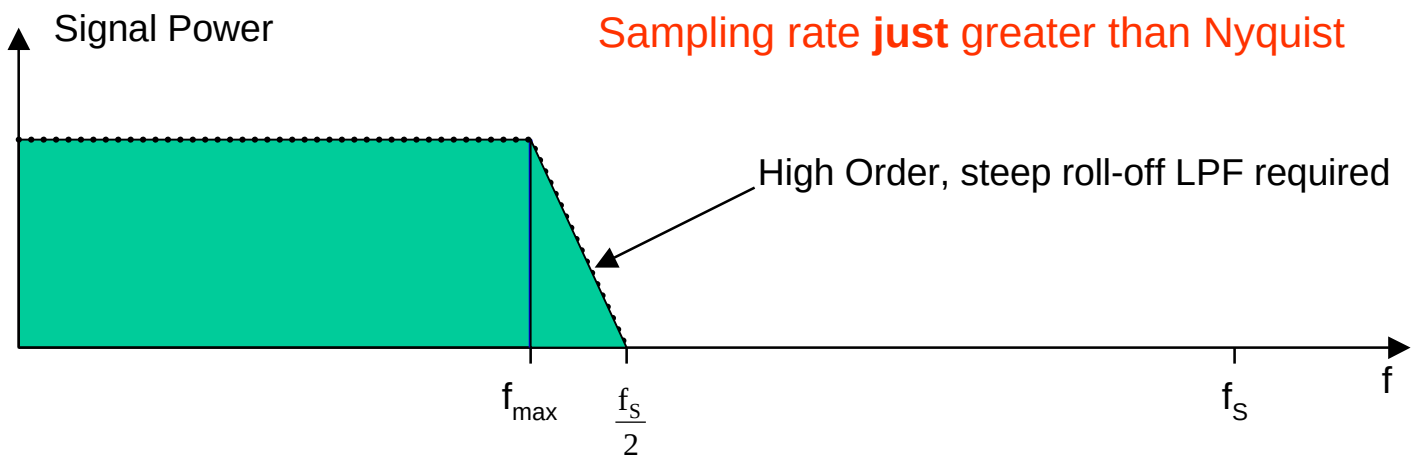


$f_S < 2f_{max}$: Aliasing



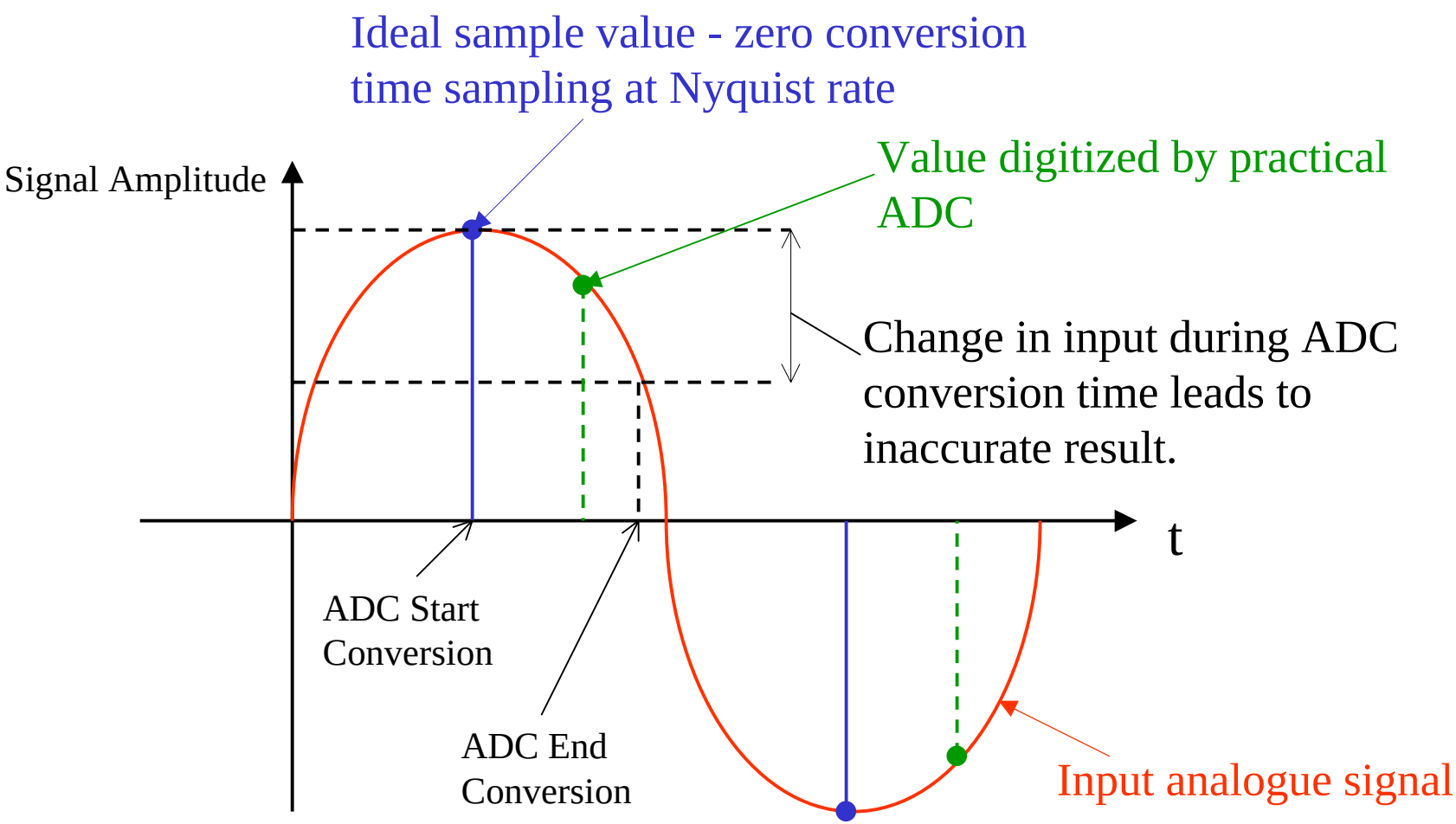
Anti-Aliasing Filter



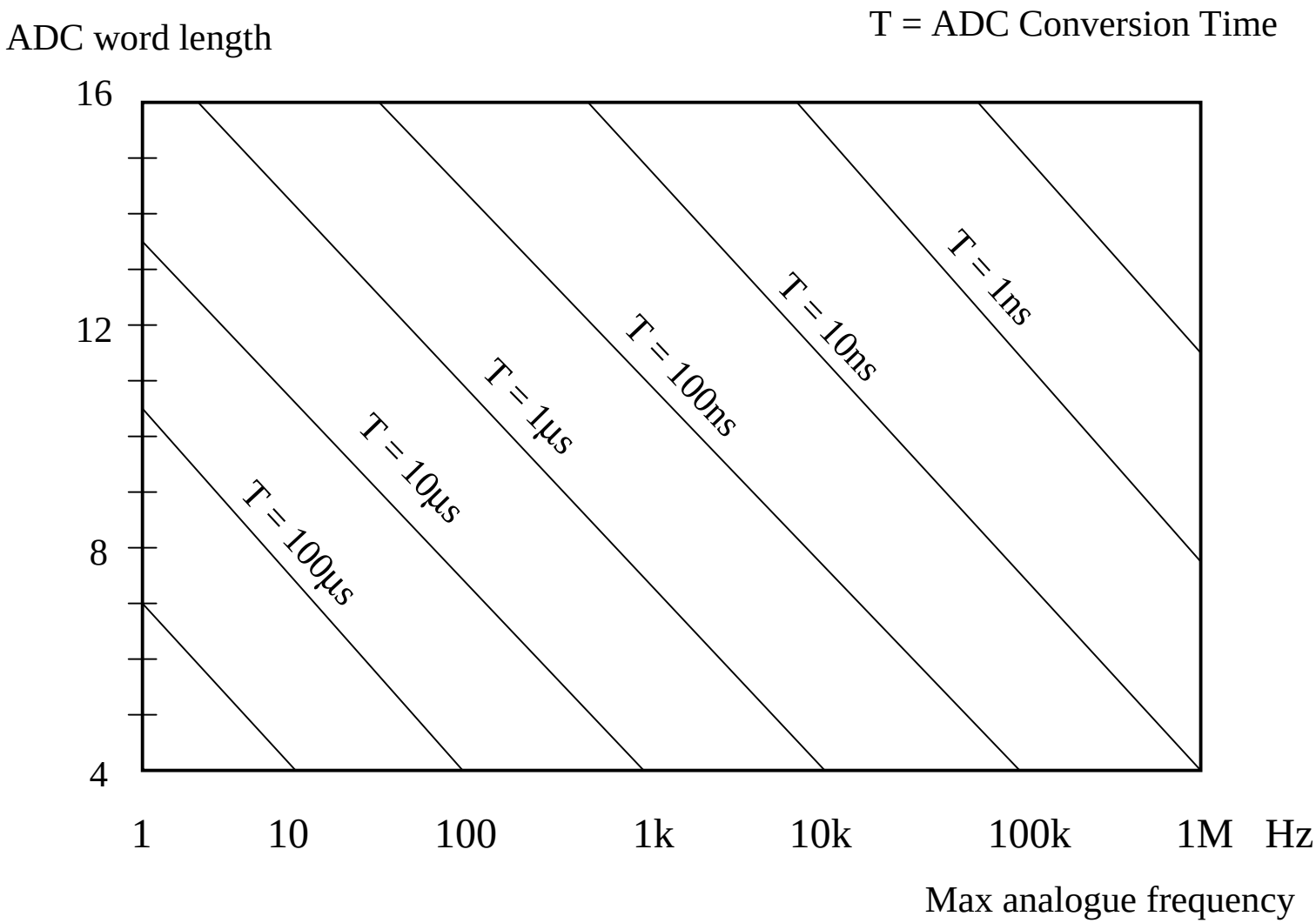


Sample-and-Hold

- Do I need one?

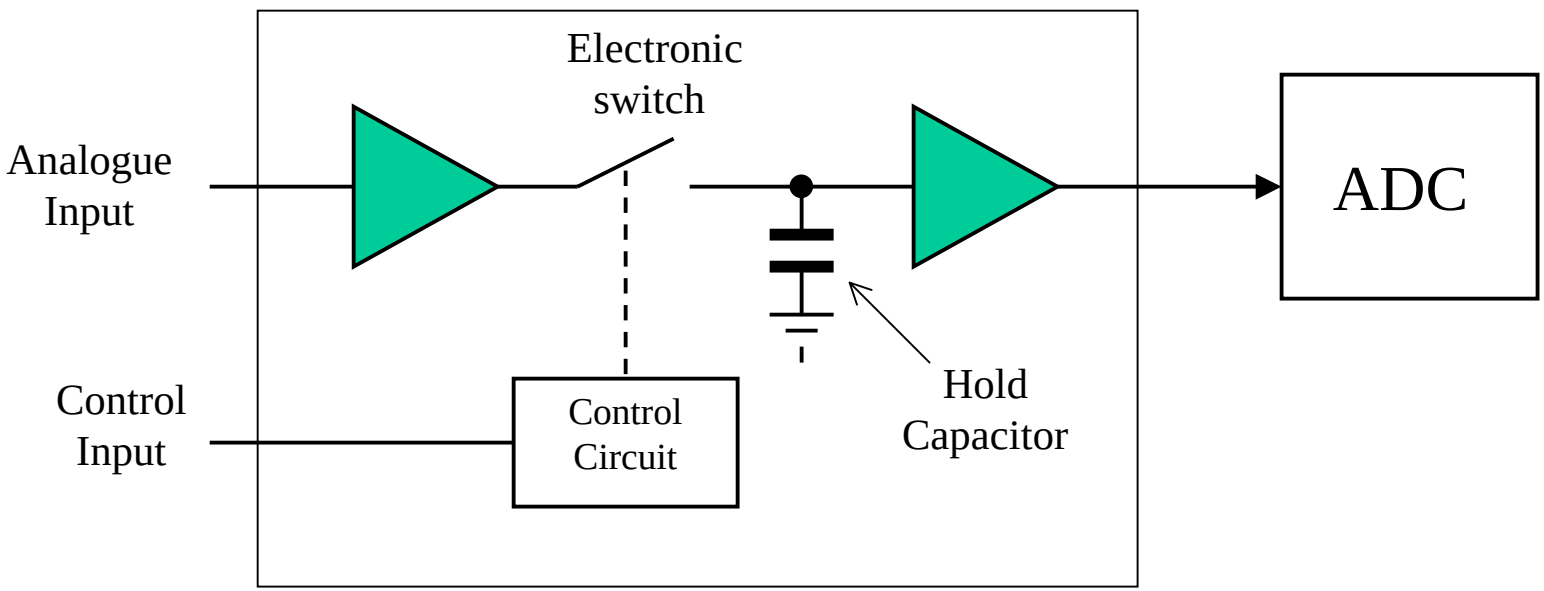


Sample-and-Hold 2



Sample-and-Hold 3

- SH unit takes a sample and holds it while the ADC converts.
- Many ADC chips contain SH built-in.



Sample-and-Hold 4

- Acquisition Time
  - Time from Hold to Tracking
- Aperture Time
  - Time from Tracking to Hold
- Voltage Droop
  - Output drop while in Hold mode
- Feedthrough
  - Input signal appearing at output while in Hold mode

ADC Type Selection

- Dual-Ramp
  - Slow, noise-insensitive. Conv time in msecs.
- Successive Approximation
  - General purpose, Conv time 30 to 100 μs
- Flash
  - Very fast, expensive, Conv time < 20 ns
- Data Acquisition Chips
  - Usually include analogue multiplexers, Dual-port RAM, DAC. Easy interface to DSPs.

ADC Size Selection

- Dynamic Range
  - Ratio of largest to smallest input signal
  - DR = 6.02n dB where n = no of bits
  - e.g. n =8 DR = 48.1 dB Accuracy = 0.4 %
  - Don't use more bits than you need
- Signal to Quantization Noise Ratio
  - SNR = 6.02n + 1.76 dB for ideal ADC
  - SNR = 6.02n - 4.24 dB worst case with ½ LSB linearity error

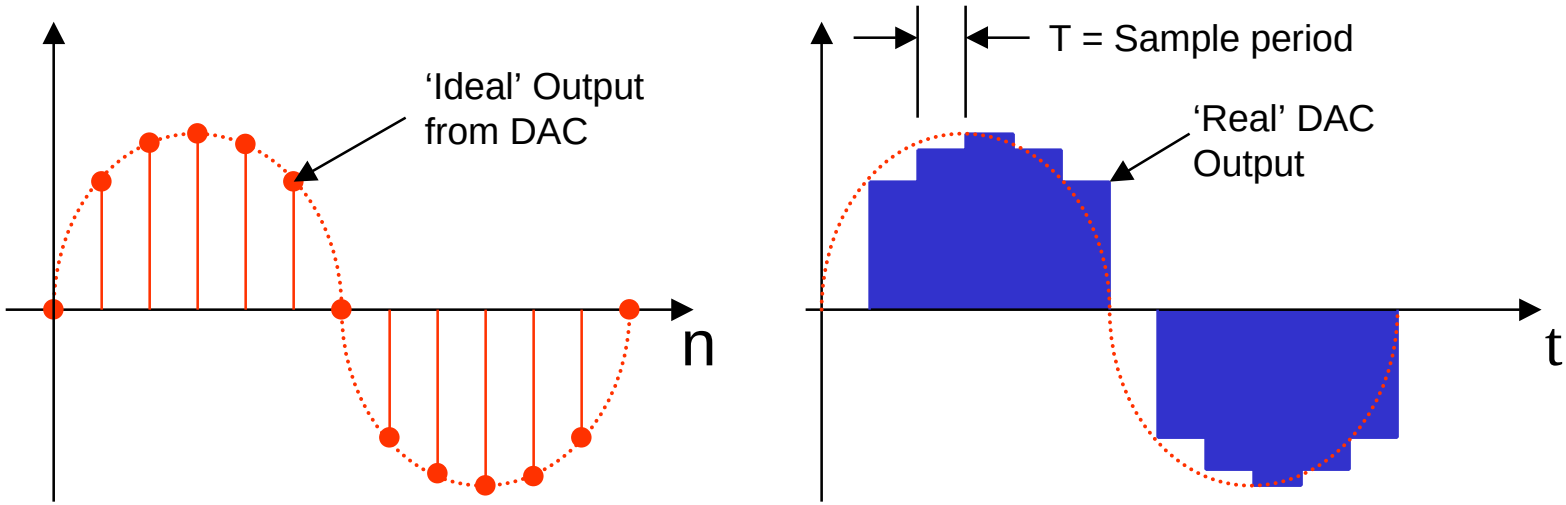
Digital to Analogue Hardware

- Full-scale output
  - Output current/voltage for all 1's input
- Resolution
- Settling Time
  - Time taken for output to settle within ½ LSB after input change.
- Linearity
- S/H used for 'De-Glitching' (Optional)
- Reconstruction filter

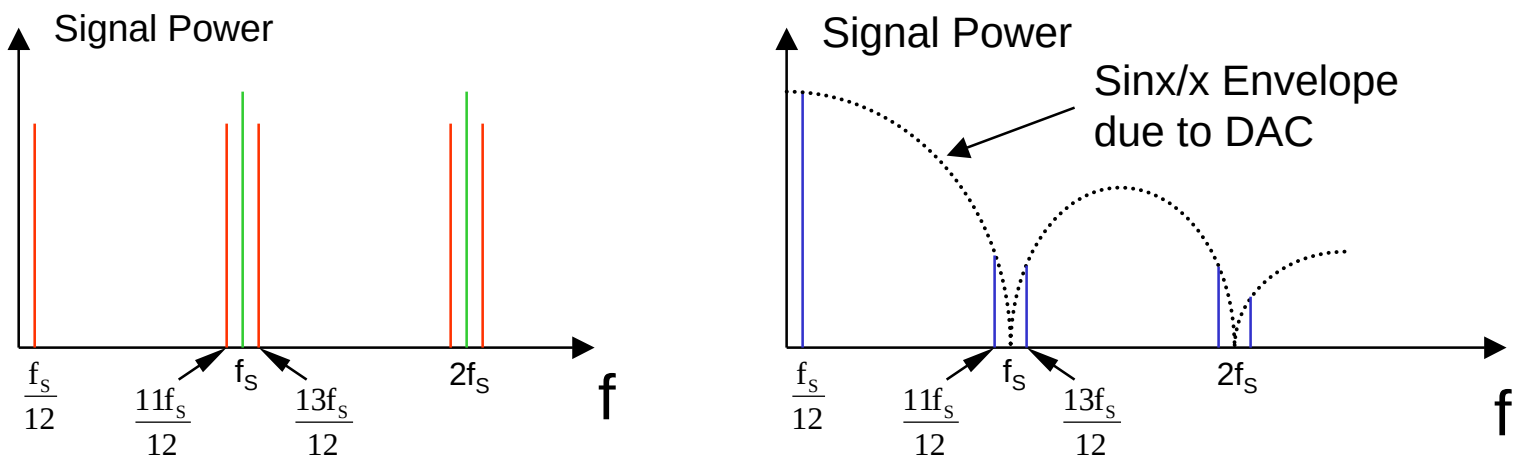
DAC Selection

- Usually matched to ADC specification
  - Usually same word-size as ADC
  - Settling Time < Sample Interval
  - Simpler and cheaper than ADC
  - Bipolar conversion required?
- DSP Interface
  - Device matched to DSP chip or more support devices required? e.g. Input latches
  - Sinx/x distortion correction

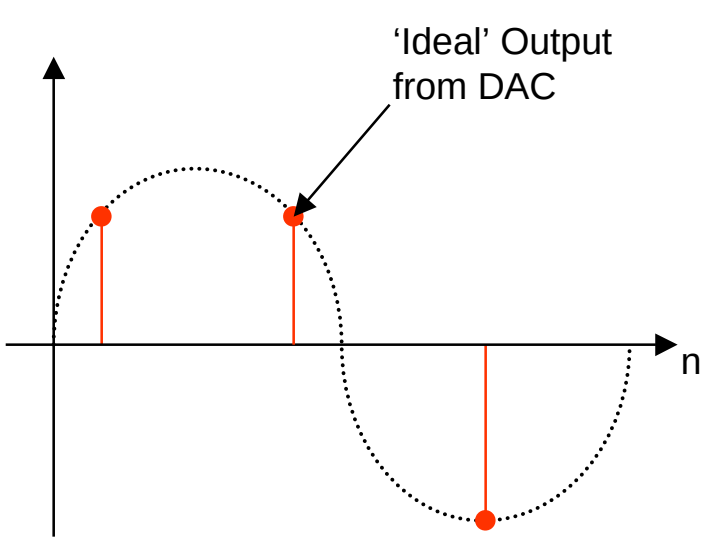
The DAC: Effect on output Spectrum

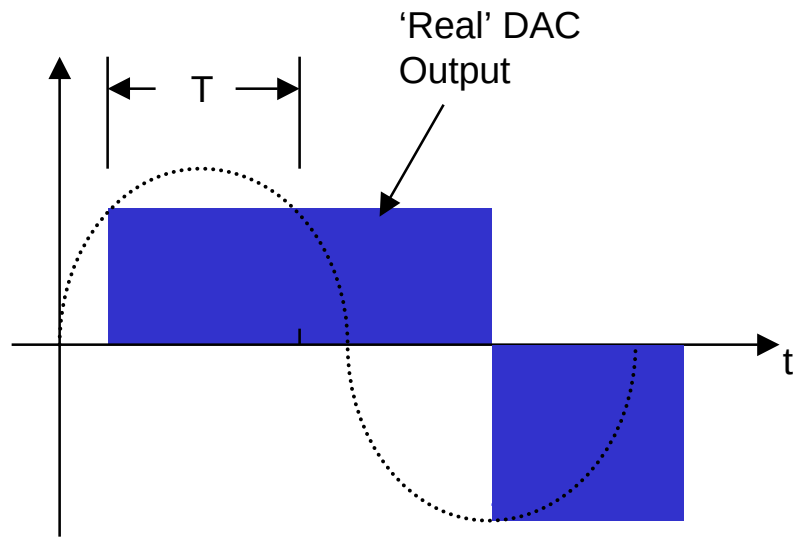


Sine wave period = 12T $f_S = 1/T$ Sine wave frequency = $f_S/12$

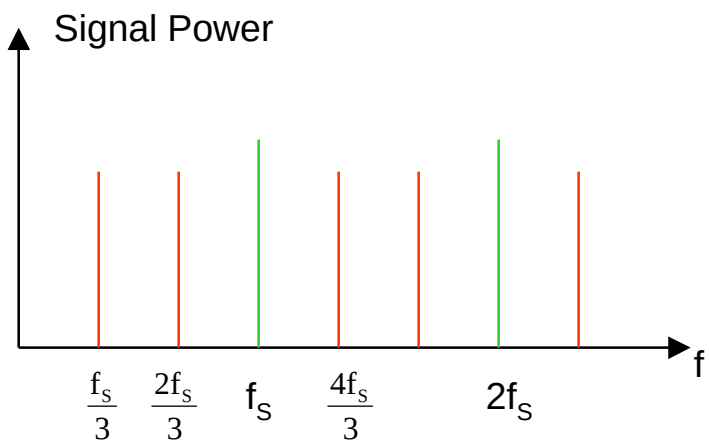


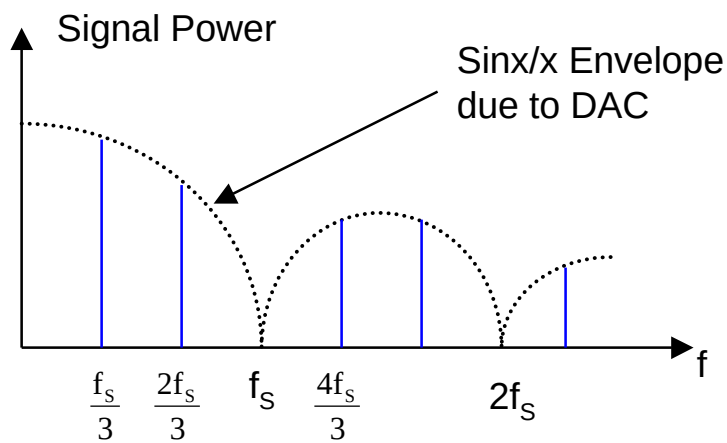
The DAC: Lower Sampling Rate



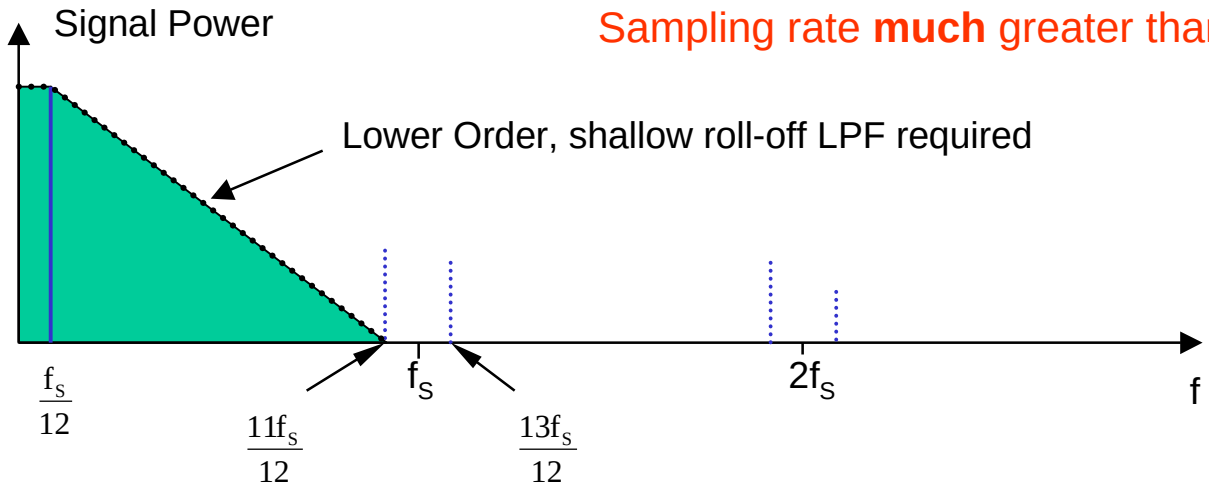


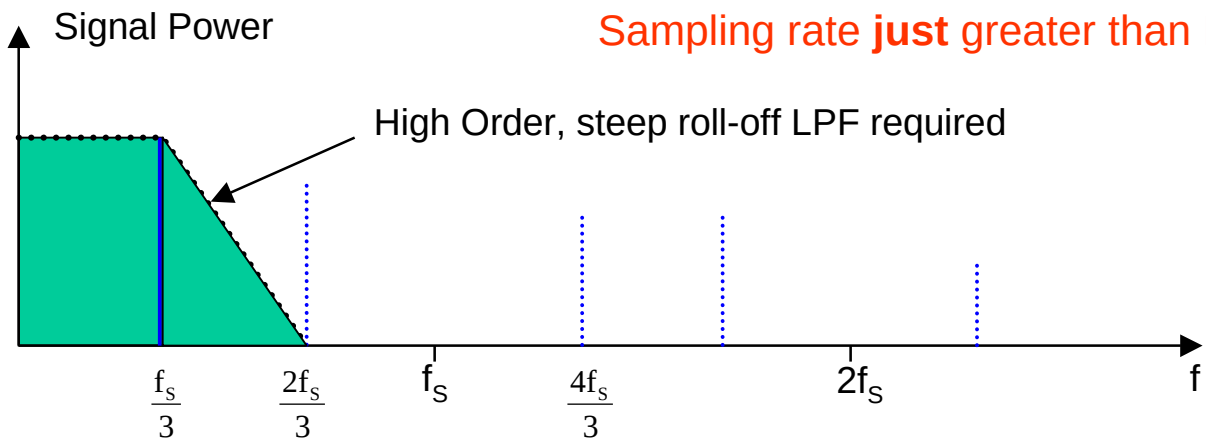
Sine wave period = 3T $f_S = 1/T$ Sine wave frequency = $f_S/3$





The DAC: Reconstruction Filter





Digital Processing Time

- Telephone channel
  - Bandwidth: 350 Hz - 3.5 kHz
  - Sampling Rate: 8 ksamples/s
  - Processing time available = 1/8000 = 125 μs
  - Typical tasks: Echo cancellation, filtering
- Stereo music
  - Bandwidth: 50 Hz - 20 kHz
  - Sampling Rate: 44.1 ksamples/s per channel
  - Processing time available = 22.7 μs per channel
  - Typical tasks: Filtering, spectral analysis

Digital Processing Time 2

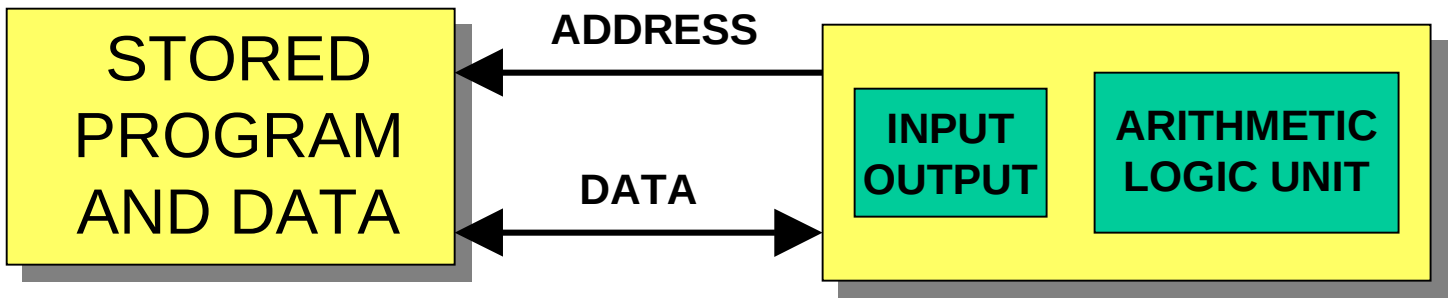
- Heart signal (ECG) monitoring
  - Bandwidth: 0.07 Hz - 100 Hz
  - Sampling Rate: 200 samples/s
  - Processing time available = 1/200 = 5 ms
  - Typical tasks: noise removal, spectral analysis
- Engine Knock Detection
  - Bandwidth: 5 kHz - 20 kHz
  - Sampling Rate: 40 ksamples/s
  - Processing time available = 25 μs
  - Typical tasks: spectral signature detection

DSP v Microprocessor

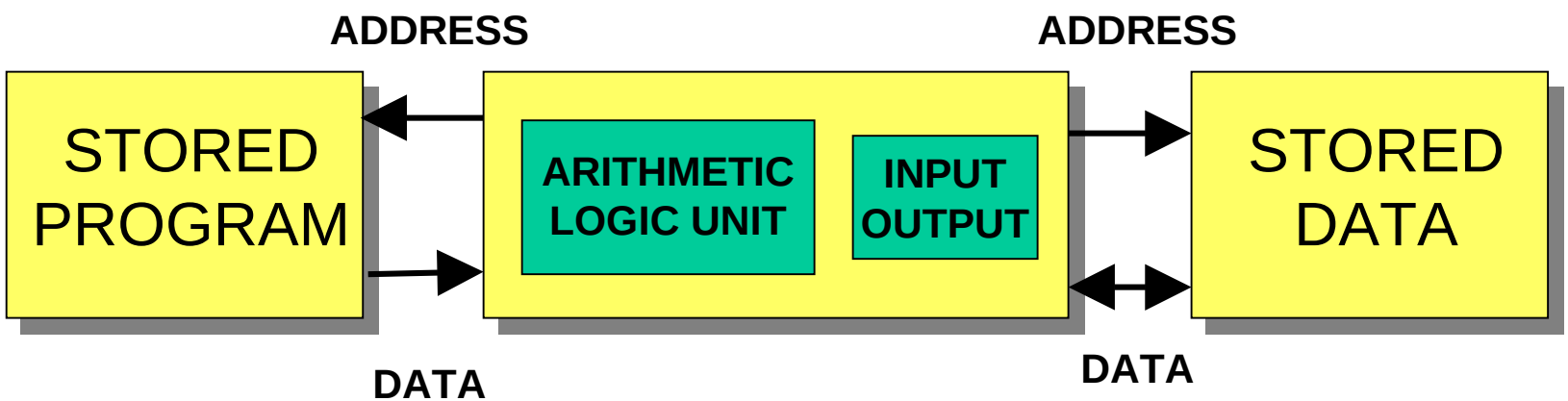
- DSP has fewer instructions (RISC)
- DSP has fewer addressing modes
- DSP instructions fast, many single-cycle
- DSP has speed-optimized instructions
  - Multiply
  - Multiply-Accumulate
- DSP has a zero-overhead loop counter
- DSP has a ‘circular’ addressing mode
- DSP needs very fast external memory

Computer Architecture

Von Neuman

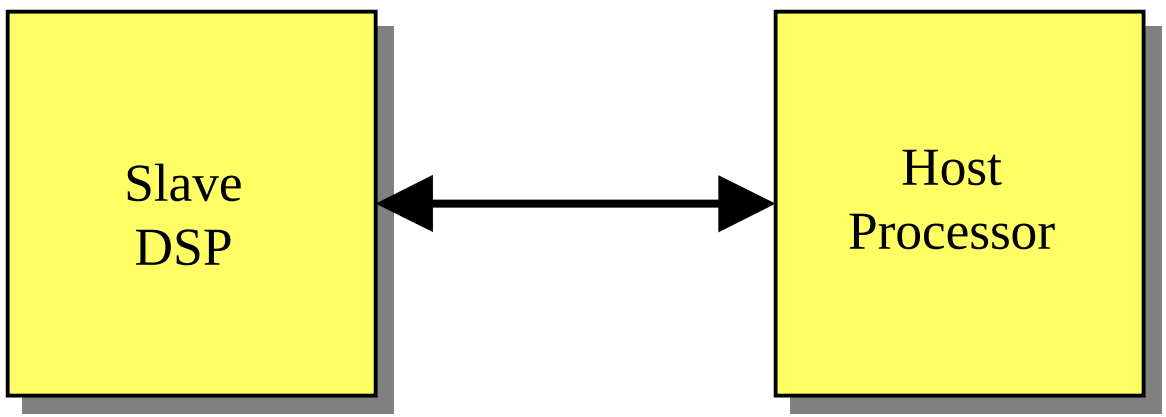


Havard



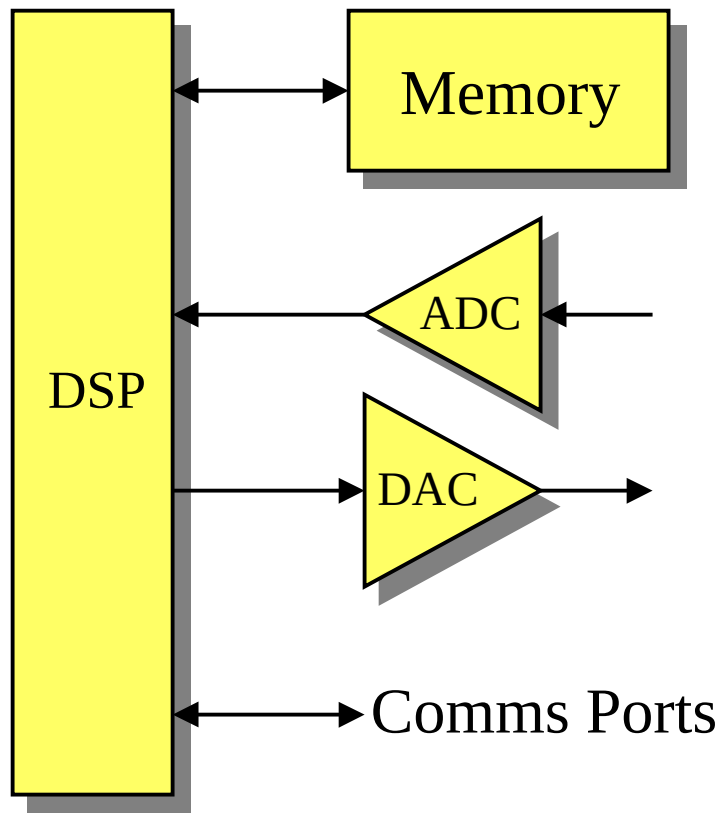
DSP as Accelerator

- A DSP cab be added as a slave to speed up calculations, but system does not input/output data in real time.

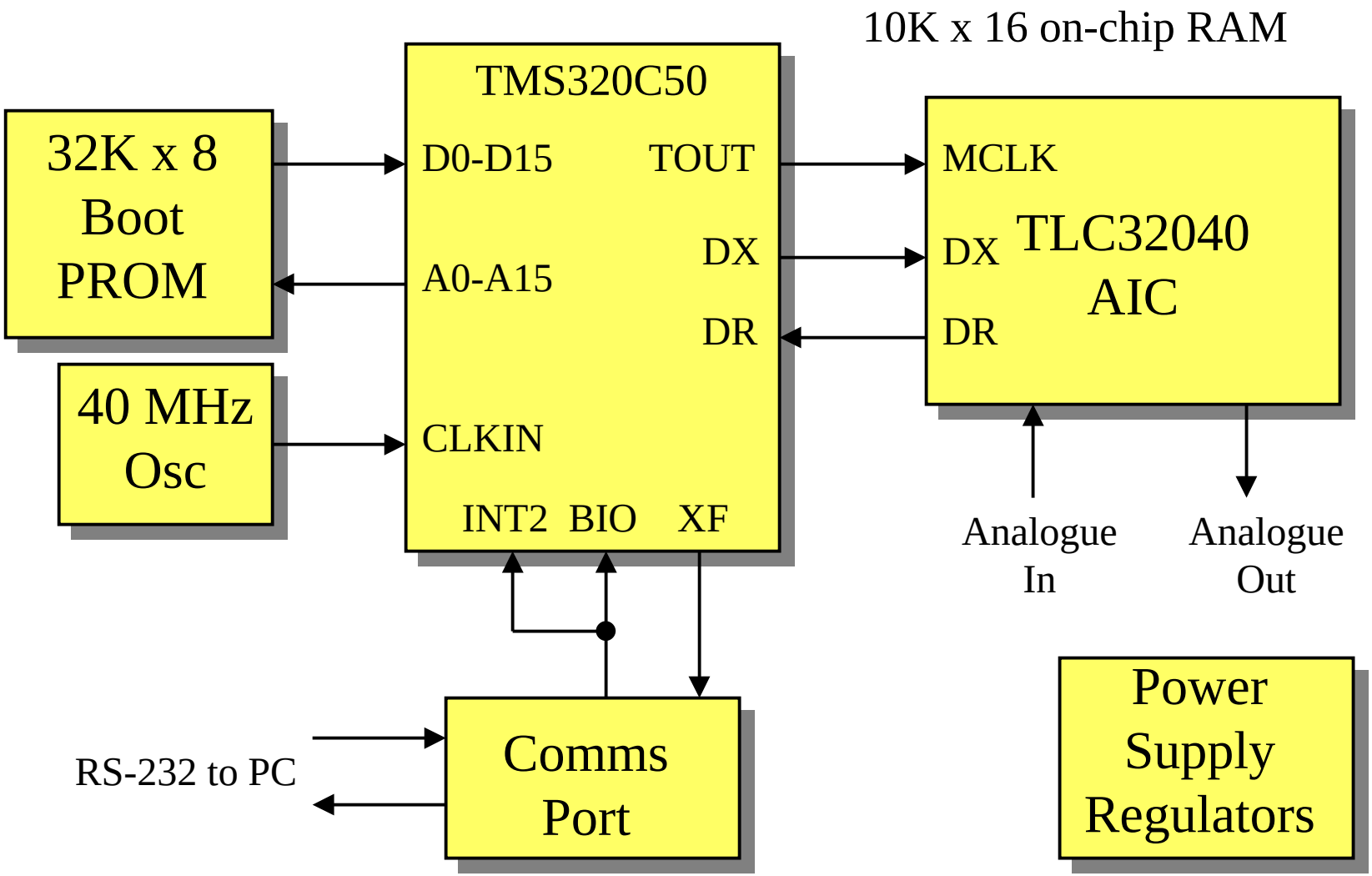


A ‘Real-Time’ DSP System

- DSP chip
- Memory chip(s)
  - Data
  - Program
- A to D Converter(s)
- D to A Converter(s)
- Comms Port(s)
  - Serial
  - Parallel



TMS320C50 DSK Hardware



Analogue Interface Circuit

- The AIC is a CODEC (COder-DECoder)
  - AIC converts sampled analogue signal to serial data (and back)
  - Max sampling frequency f_S = 19200 samples/s
  - 14 bit ADC/DAC resolution
- Programmable anti-alias filter
- Programmable reconstruction filter
- Designed to interface to DSP serial port

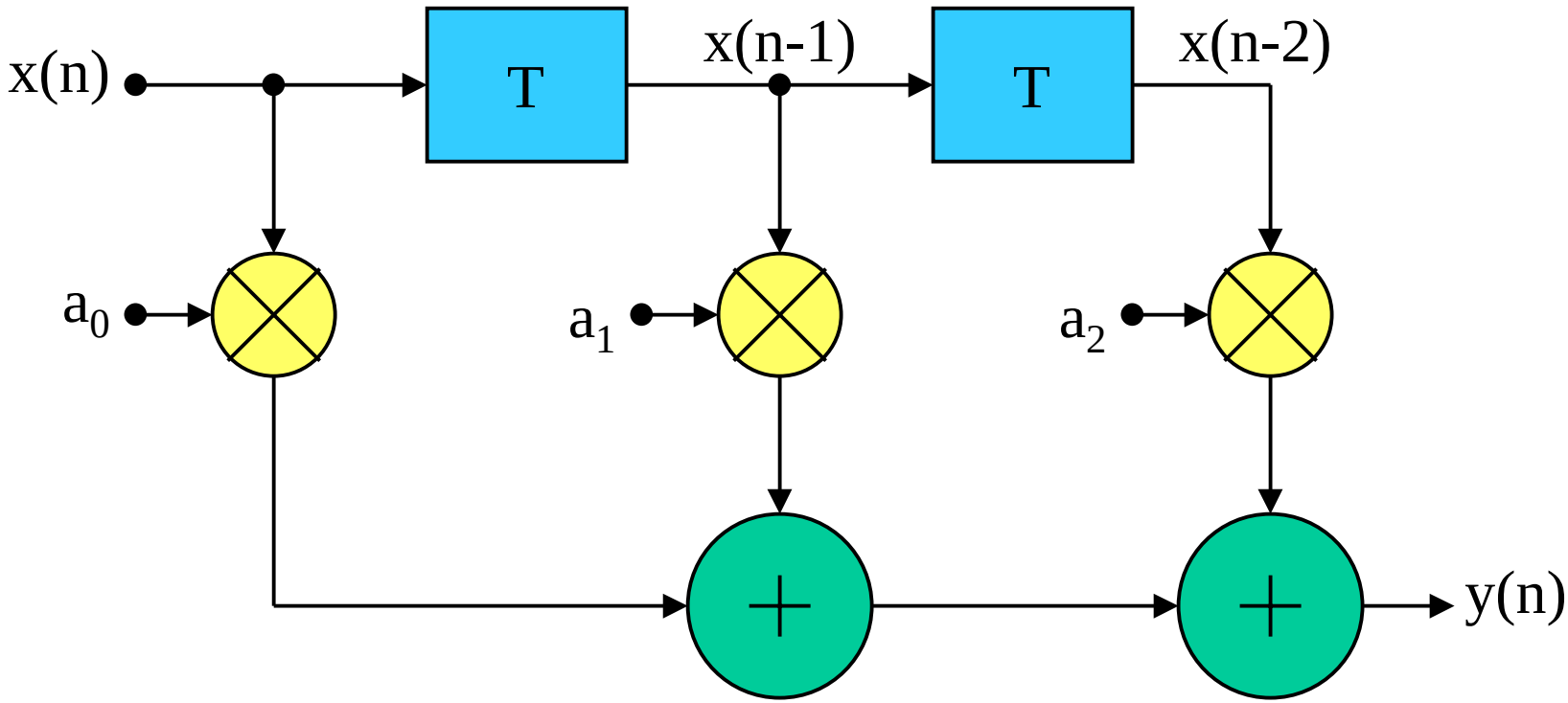
Program Development

- Generate assembler code file (*.asm)
  - convert algorithm
  - add initialisation code to suit DSK system
  - add input/output code to suit ADC/DAC
- Generate machine code file (*.dsk)
  - using DSK5A assembler
- Download machine code to DSK and run
  - using DSK5L loader
- Debug using DSK5D debug tool

Sample and Block Processing

- Sample Processing
  - A processed output sample is produced every time a new input sample is taken.
  - Algorithm completed during a sample period.
  - e.g. FIR filter
- Block Processing
  - A set of samples taken, *then* processed. A *previous input set* is processed while the new samples come in, and *previous* results go out.
  - e.g. Fast Fourier Transform

Sample Processing - FIR Filter



a_0 , a_1, a_2 = Multipliers or 'Tap Gains' T = Sample Period = $1/f_S$

$$y(n) = a_0 \times x(n) + a_1 \times x(n\text{-}1) + a_2 \times x(n\text{-}2)$$

The FIR Filter

- FIR = Finite Impulse Response
  - Tap Gains are a truncated (windowed) set of filter impulse response samples.
  - Only past and current input data samples used.
  - Always stable
  - Perfect linear phase response
- Filter Order = No. of previous inputs used
  - Order of example on previous slide = ?
- Higher order for same roll-off than IIR filter
- Need design software to calculate tap gains

The FIR Filter 2

- Decide on Sampling rate f_S
- Decide on filter cut-off frequency, roll-off, stop-band attenuation
- Use design software e.g. MatLab® to get Tap Gain set
- Scale Tap Gains for signed 16-bit fixed-point arithmetic (Q15)
- Program FIR filter in DSP code with tap gains in ‘look-up’ table

Fixed versus Floating Point

- TMS 320C50 is a Fixed Point DSP
  - Stored numbers are 16 bits wide
  - Accumulator is 32 bits wide
  - Integer number Range: -32768 to +32767
  - Overflows in intermediate calculations can occur
- TMS320C40 is a Floating Point DSP
  - Number range: $+3.4\times10^{38}$ to -3.4×10^{38} (40 bit)
  - Numerical overflows less likely
  - Processing less efficient

Scaling Tap Gains in Fixed Point

- Design packages usually produce values in the range -1 to +1
- If max value ≠ 1, divide it and all other values by max value to normalize and maintain accuracy.
- Convert to Q15 format by multiplying by 2^{15} -1 (= 32767 or 7FFFh)
- Scaling may need adjustment if overflow occurs.

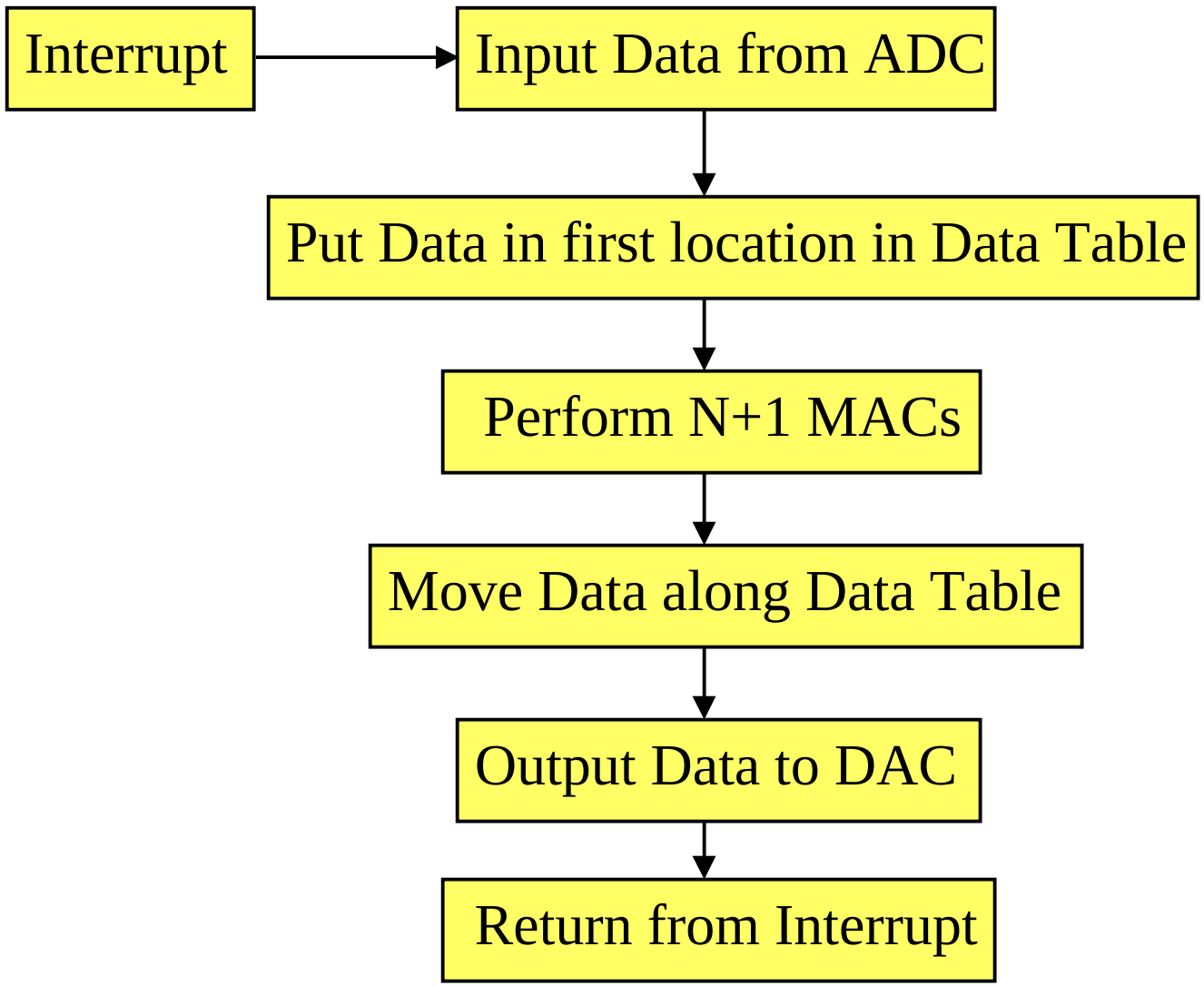
Handling Overflows

- Set Overflow Mode bit (OVM) in DSP while de-bugging, for intermediate overflows. Converts overflow to ‘clipping’.
- Test Overflow flag (OV) in software for non-intermediate overflows
- ADC data usually less than 16-bit, reducing likelihood of overflows. (DSK =14 bit)
- Scale tap gain values down if problem persists (or reduce analogue input).

FIR Program Construction

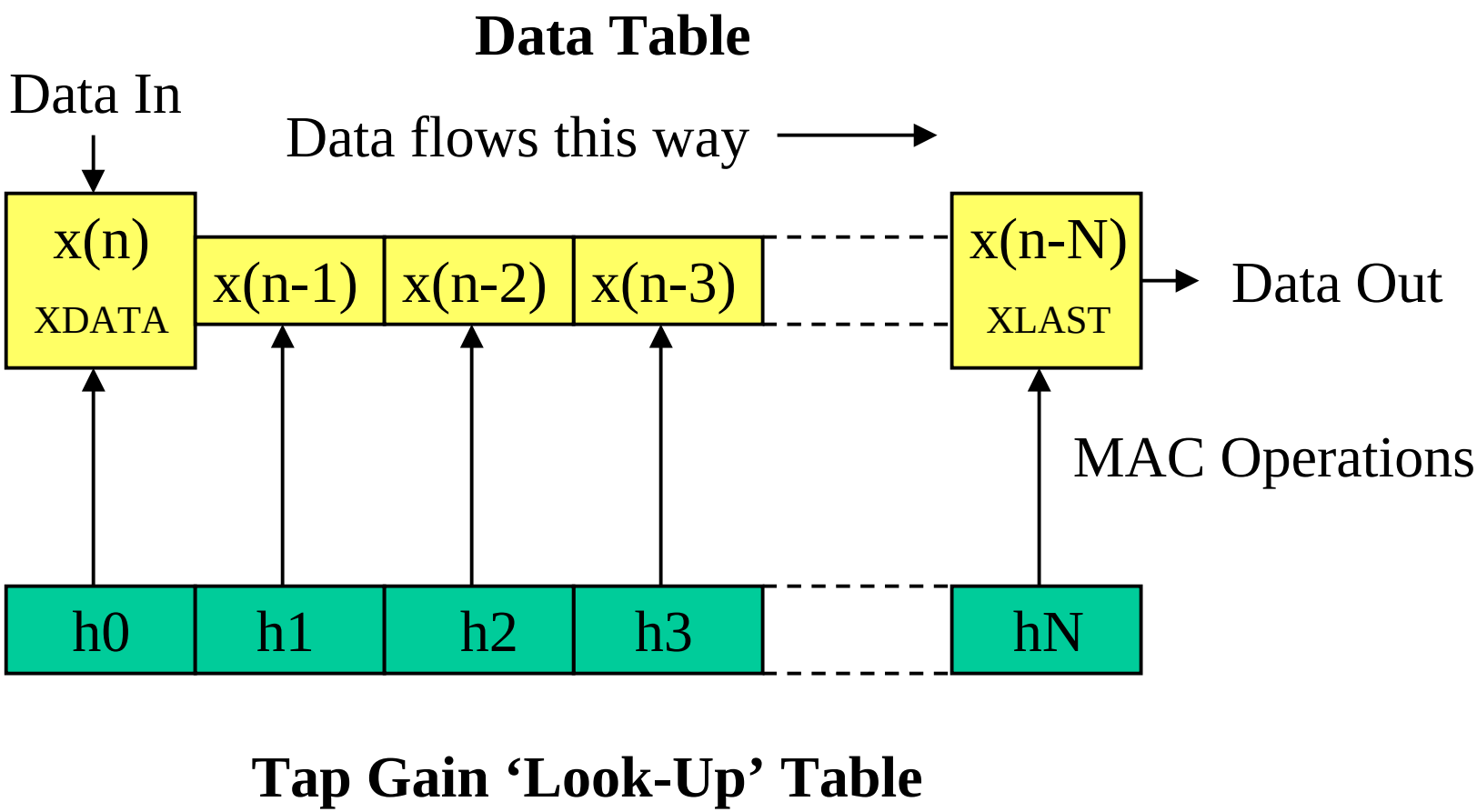
- **Base Program**
  - Initialize everything and then loop doing nothing (IDLE) except waiting for interrupt.
- **Interrupt Service Routine (ISR)**
  - Does all the work
  - Interrupt signals new sample from ADC, and processed data sample output to DAC.
  - Interrupt interval = Sampling interval
  - For filter order N, then N+1 multiply then add operations required - *between interrupts*.

ISR Operation



ISR Operation 2

Tables in DSP Memory



FIR Core Program (‘C50)

```
ldp     #0          ; load data page 0
lamm    DRR         ; read serial input (from ADC)
and     #0FFFCh     ; remove control bits
ldp     #XDATA      ; load XDATA data page
sacl    XDATA       ; received word to XDATA
lar     AR0, #XLAST ; AR0 = last delay element
zap                 ; clear acc and product register
mar     * , AR0     ; Make AR0 current pointer
rpt     #79         ; repeat next line 80 times
macd    #h0 , * -   ; do 80 MACs and data moves
apac                ; add final product to acc
sach    OUTPUT , 1  ; remove surplus sign bit
```

Block Processing - DFT

- Discrete Fourier Series
  - Finds spectral components of *periodic* signal.
- Discrete Fourier Transform (DFT)
  - Most signals encountered *not* periodic.
  - DFT finds N spectral components of an aperiodic signal of N samples.
  - Nyquist Rate: $f_N = f_S/2$
  - Min Frequency: $f_{MIN} = f_N/N$
  - Indexed frequency: $f[k] = k \times f_{MIN}$
  - $k = 0, 1, 2, \ldots N - 1$

Using the DFT

- DFT algorithm produces one complex frequency component X(k) from N signal samples. It requires 2N multiply-accumulates.
- Use when only a few spectral lines needed.
- x(n) is usually real (not complex), hence:
- Requires $2N^2$ MACs for all N components.
- For a given f_S, increasing N increases the frequency resolution.
- N can be any number (not just power of 2)

Practical DFT

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{\frac{-j2\pi kn}{N}} = \sum_{n=0}^{N-1} x[n]\, W_N^{kn}$$

where k = 0, 1, ….N-1 and W_N = Phase factor

Difficult to compute complex exponential format on a fixed-point DSP, so substitute:

$e^{-j\theta} = \cos\theta - j\sin\theta$ which yields:

$$X[k] = \sum_{n=0}^{N-1} x(n)\cos(2\pi kn/N) - j\sum_{n=0}^{N-1} x(n)\sin(2\pi kn/N)$$

Practical DFT 2

Separating Real and Imaginary Parts:

$$X_R[k] = \sum_{n=0}^{N-1} x(n) \cos(2\pi kn / N)$$

$$X_I[k] = \sum_{n=0}^{N-1} x(n) \sin(2\pi kn / N)$$

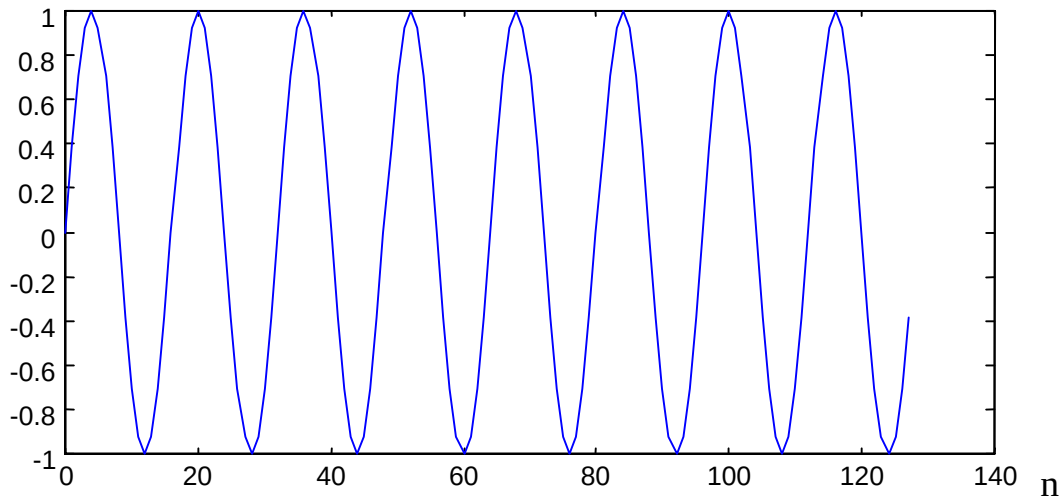
Magnitude: $$|X[k]| = \sqrt{X_R[k]^2 + X_I[k]^2}$$

Phase: $$\phi[k] = \arctan \frac{X_I[k]}{X_R[k]}$$

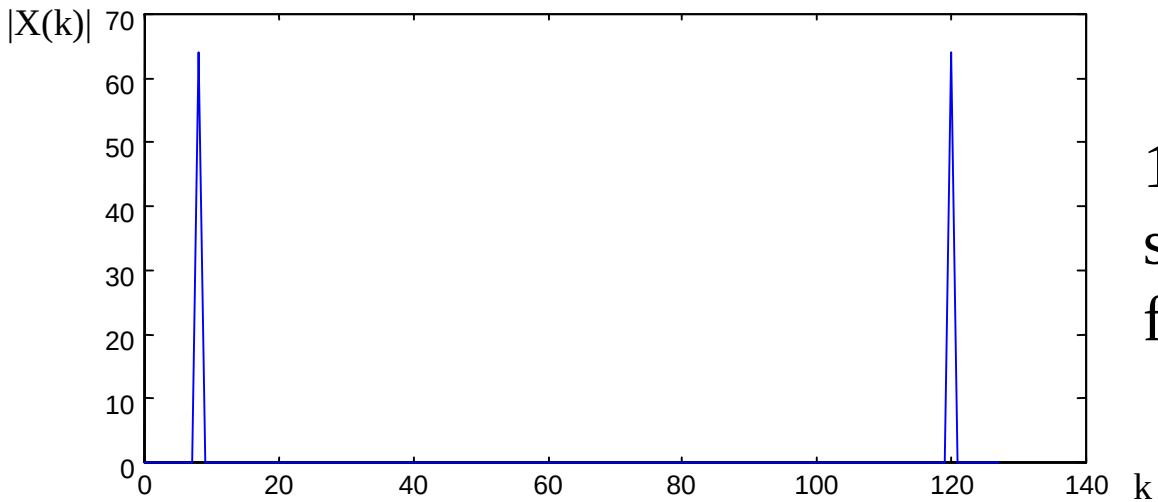
Practical DFT 3

- Use pre-calculated look-up tables for sine and cosine functions.
  - e.g. If N = 8 then k = 0….7 and n = 0….7
    Hence 64 sine and 64 cosine table entries.
  - Many entries are equal, 0, 1 or negatives
- Often don't need the phase information
- Often *relative* magnitude only required so don't bother with square root.
- |X[0]| to |X[N/2]| mirror of N/2 to N-1

DFT Example 1

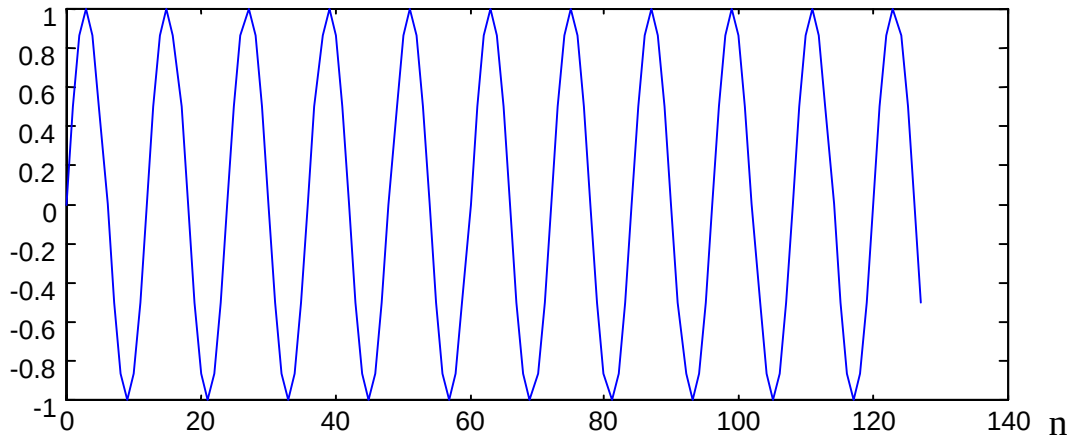


N=128 8 cycles of sine wave at $f_S/16$

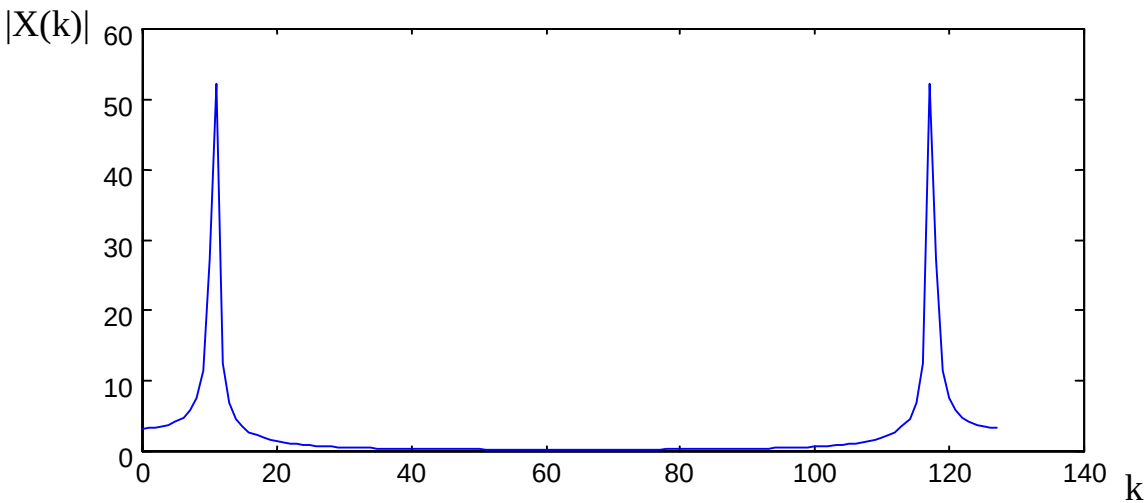


128-point DFT shows single lines at $f_S/16$ and 128 - $f_s/16$

DFT Example 2 - Spectral Leakage



N=128 11.1 cycles of sine wave at $f_S/11.53$



128-point DFT shows peaks (Multiple lines) around $f_S/11.53$ and $128 - f_s/11.53$

Faster DFT - The FFT

- DFT heavy on processor power
- Many phase factors identical
- Many duplicated multiplications
- Re-arrange order of processing to remove duplication
- DFT processing requirement α N^2
- FFT processing requirement α $N\log_2 N$
- Improvement increases with increasing N

FFT 2

- Decompose DFT length N into two DFTs length N/2.
- One part uses even-numbered n of x[n]
- Other part uses odd-numbered n of x[n]
- If n = 2r for even and n = 2r+1 for odd then:

$$X[k] = \sum_{r=0}^{N/2-1} x[2r]\, W_{N/2}^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1]\, W_{N/2}^{rk}$$

$$= G[k] + W_N^k\, H[k]$$

FFT 3

- N must be a power of 2. e.g. 128, 256, etc.
- The decomposition may be continued to yield a number of 2-point DFTs.
- This is called a Radix 2 Decimation in Time (DIT) FFT.
- e.g. Sampled signal x(n) length N = 8
  - n = [0 1 2 3 4 5 6 7]
  - First decomposition: n = [0 2 4 6] [1 3 5 7]
  - 2nd decomposition: n = [0 4] [2 6] [1 5] [3 7]

FFT 4

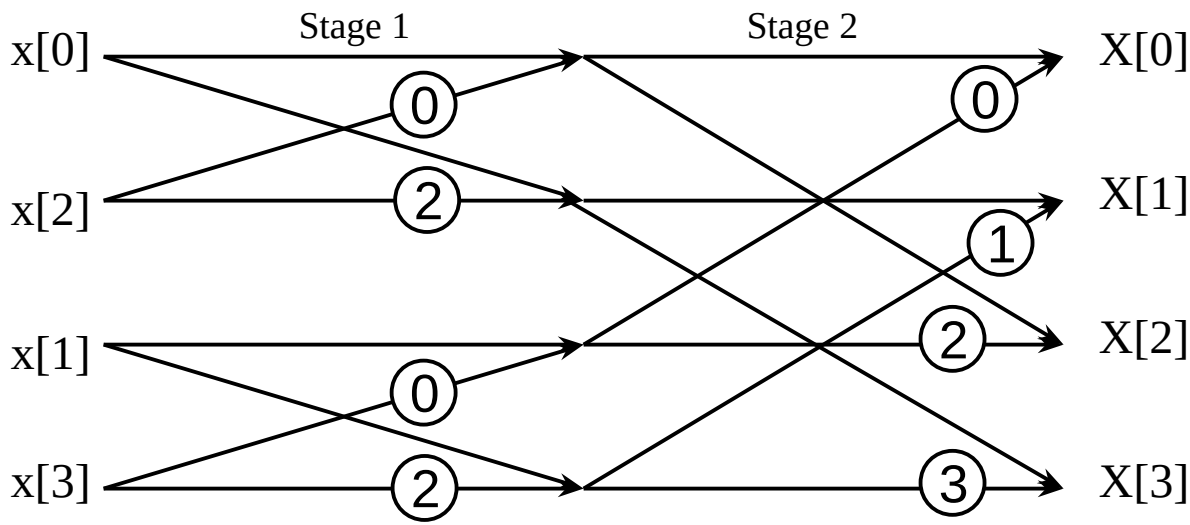
- e.g. Equations for a 4-point FFT (N = 4)

$$X[0] = \{x[0] + x[2]W_4^0\} + W_4^0\{x[1] + x[3]\,W_4^0\}$$

$$X[1] = \{x[0] + x[2]W_4^2\} + W_4^1\{x[1] + x[3]\,W_4^2\}$$

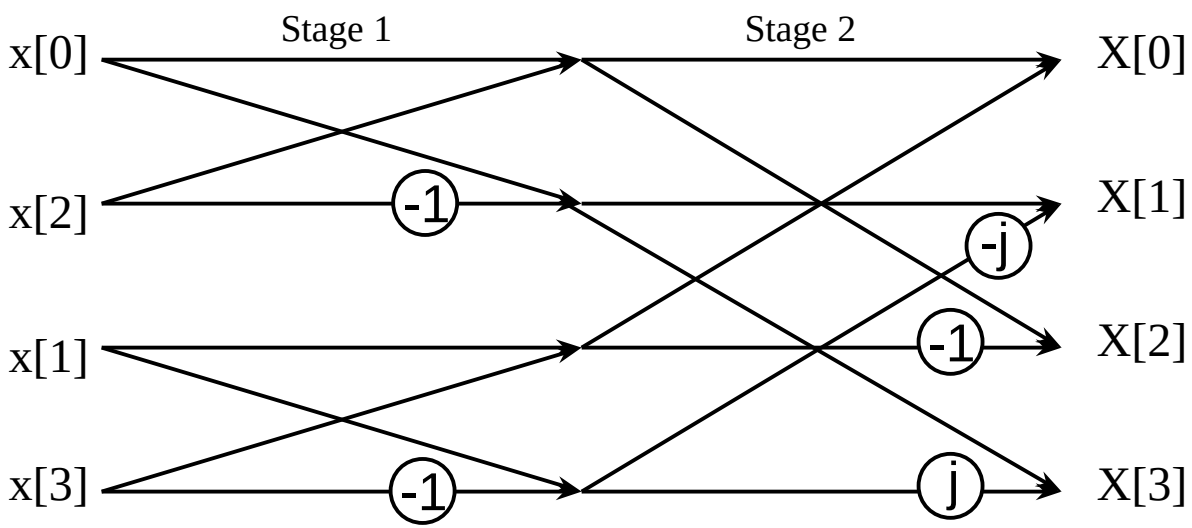
$$X[2] = \{x[0] + x[2]W_4^0\} + W_4^2\{x[1] + x[3]\,W_4^0\}$$

$$X[3] = \{x[0] + x[2]W_4^2\} + W_4^3\{x[1] + x[3]\,W_4^2\}$$



FFT 5

- Calculate W_4 factors
  - If N = 4: $W_4^0 = 1$ $W_4^1 = -j$ $W_4^2 = -1$ $W_4^3 = j$
  - Note ±j means convert between real and imaginary, -1 is simply a negation.
  - Hence:



Programming the FFT

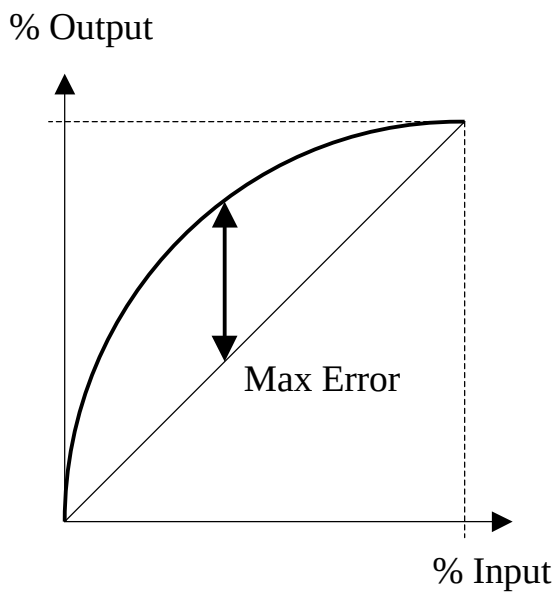
- No. of stages = $\log_2 N$
  - Program usually uses 3 nested loops: outer loop is stage counter, inner loops compute ‘butterflies’.
  - N/2 butterflies per stage.
- Butterflies computed ‘in-place’.
- Data ‘shuffled’:
  - First (Decimation in Time) or
  - Last (Decimation in Frequency) or
  - Use special addressing mode on ‘C50 chip.
- Each stage output scaled by 2 (overflows)

Sensor Specifications

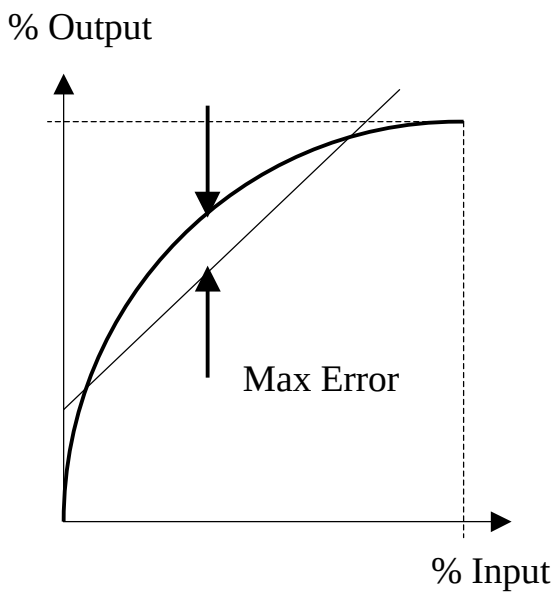
- **Range**: limits between which inputs can vary.
- **Error**: measured value - true value.
- **Accuracy**: Likely variation from true value.
  - e.g. Specification: ±5% of Full Scale Output. If FSO = 100^{O}C, then reading may be ±5^{O} off true.
- **Sensitivity**: Relates output to input amplitude
  - e.g. Specification: 0.4 Ω/^{O}C
  - Can also indicate other influences, e.g. PS voltage
- **Resolution**: Smallest change in input that will produce observable change in output.

Sensor Specifications 2

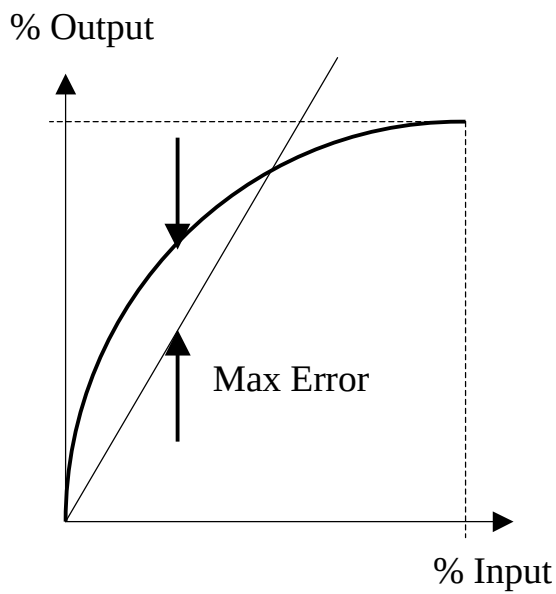
- **Non-Linearity**: Max distance from ideal linear response line. Three methods of expressing the error:



End-point values



Best line - all values



Best line - thro’ zero

Sensor Specifications 3

- **Repeatability**: Measure of device ability to provide same output with repeated applications of same input value.

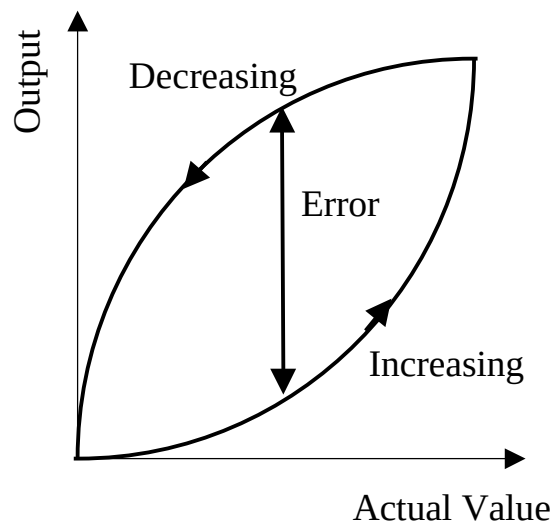
$$R = \frac{\max - \min}{\text{full range}} \times 100$$

- **Stability**: Ability to give the same output with a *constant* input over time. i.e. tendency to *drift*.

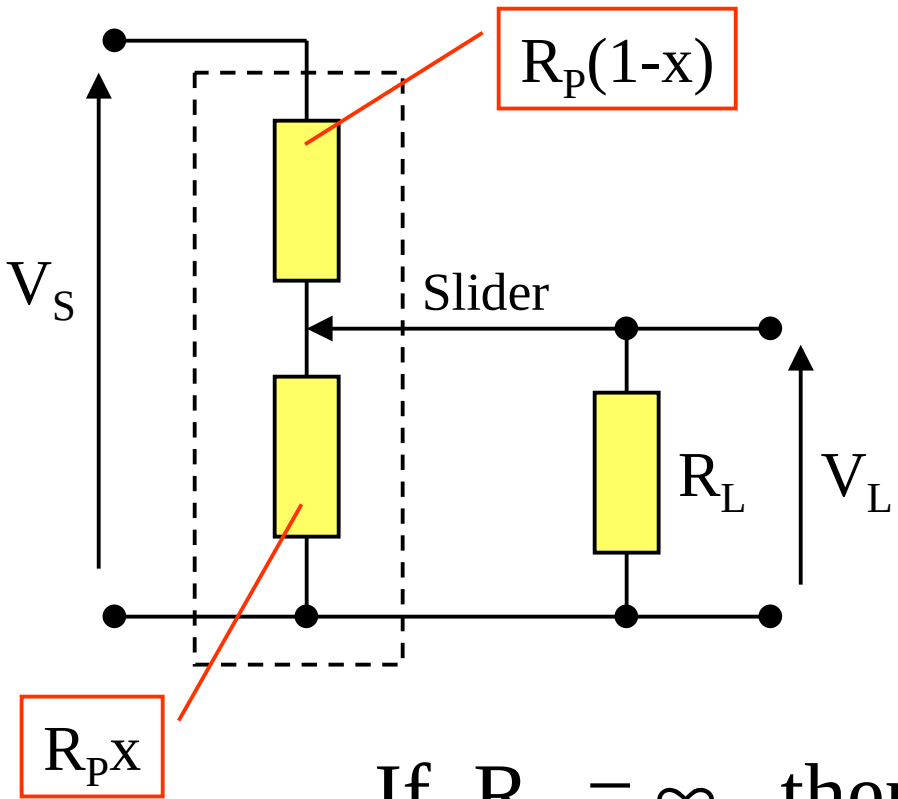
  Zero drift describes output changes when input is zero.

Sensor Specifications 4

- **Dead-Band:** Range of input values (either side of zero) for which no output.
  - e.g. Bearing friction in turbine-type transducer.
- **Output Impedance**: Often high, so high-impedance input amplifiers needed.
- **Hysteresis Error**: Different value according to rising or falling trend:



Position Sensing - Potentiometer



R_P = Pot resistance

R_L = Load resistance

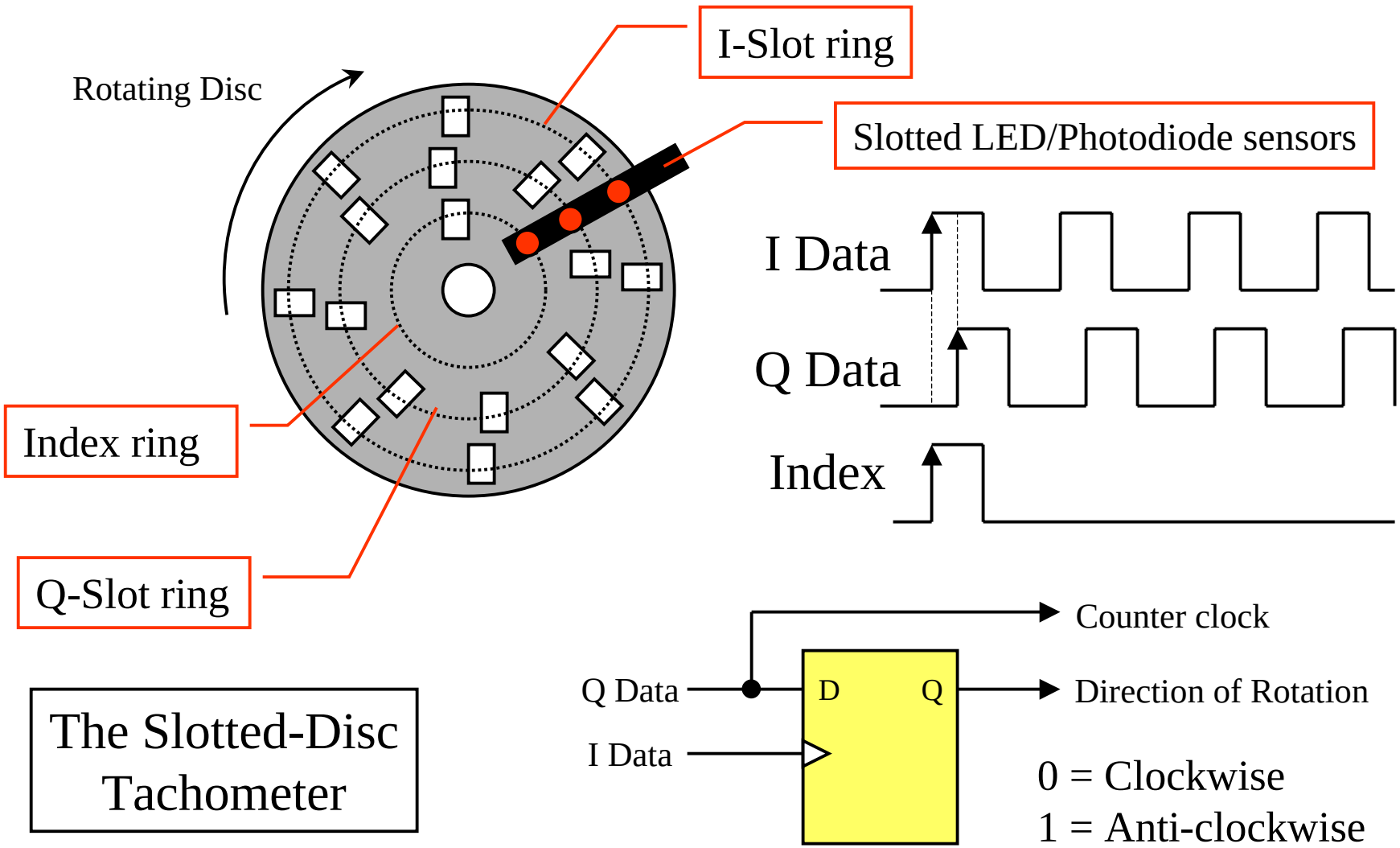
Position α V_L

Error due to Loading

If $R_L = \infty$ then $V_L = xV_S$

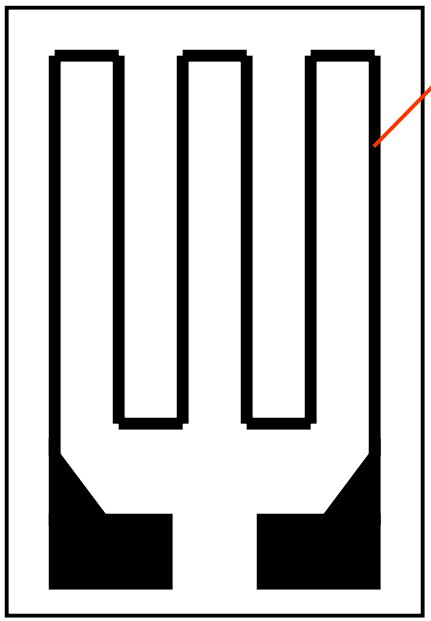
If $R_L < \infty$ then $\text{error} = \frac{R_P}{R_L}\left(x^2 - x^3\right)$

Speed/Rotation Sensing



The Slotted-Disc Tachometer

Strain Gauges



Un-strained Resistance = R ohms
Change in resistance = ΔR due to applied strain ε

$$\frac{\Delta R}{R} = G\varepsilon$$ where G = Gauge Factor

G ≈ 2.0 for metal types, G ≈ 100 for semiconductor types.
Temperature also affects both types.

e.g. Metal gauge, factor G = 2.0, un-strained resistance R = 100 Ω. What is ΔR given a strain of 0.001?
Change in resistance = $2.0 \times 0.001 \times 100 = 0.2\ \Omega$