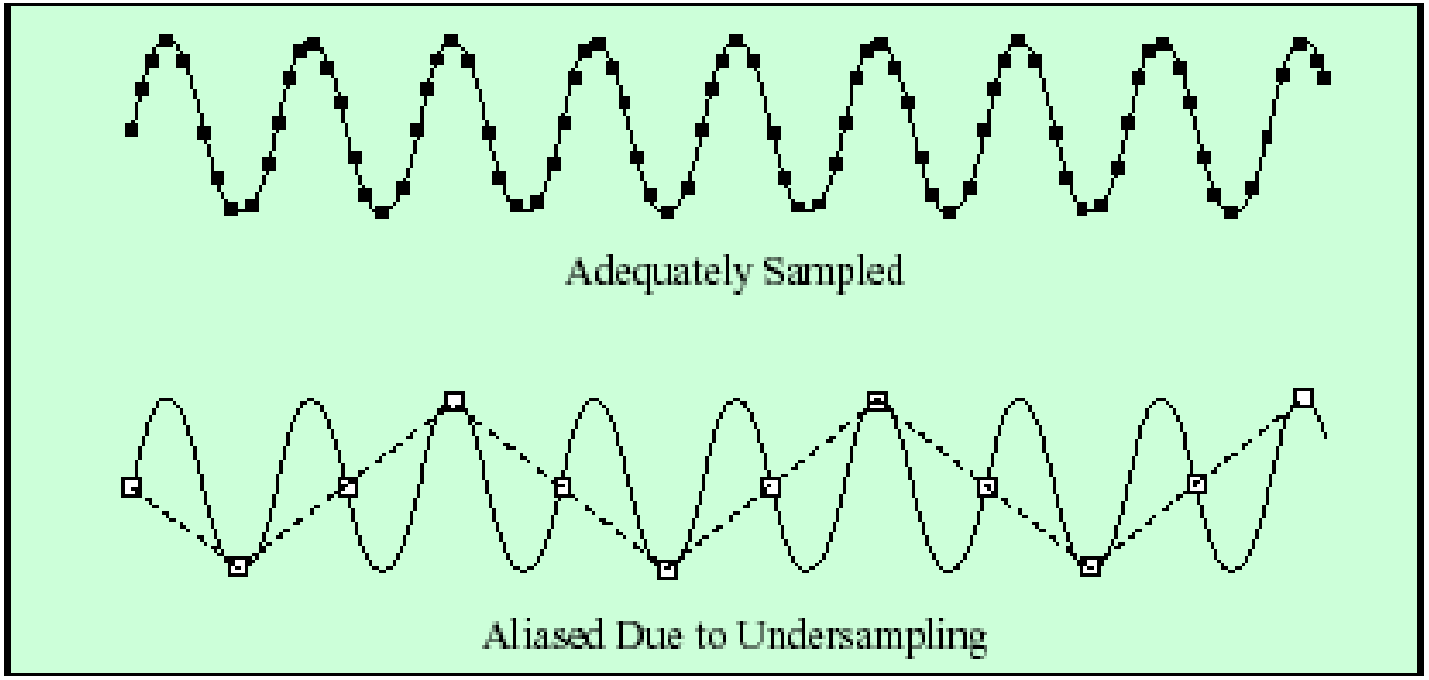
Sampling and Analogue to Digital Conversion

Shannon's Sampling Theorem



Shannon's Sampling Theorem

- A function f(t) with a bandwidth ω_b is uniquely determined by a discrete set of sample values provided that the sampling frequency is greater than $2\omega_b$.

- The frequency $2\omega_b$ is called the Nyquist frequency.

Shannon's Sampling Theorem

- A good rule of thumb is to sample the signal at about ten times ω_b.
- $\omega_s = 10\,\omega_b$
- To prevent aliasing use an anti-aliasing filter before the A/D converter. This is a L.P. filter with cut-off at $0.5\,\omega_s$.

Aliasing

- Let the sampled signal be $x_s(t) = x(t)s(t)$ where $s(t)$ is a sampling function with period f_s.
- $s(t)$ can be written as a Fourier series.

$$s(t) = c_0 + \sum_{n=1}^{\infty} 2c_n \cos n\omega_s t$$

Aliasing

- x_s can then be written as an expansion:

$$x_s(t) = c_0 x(t) + 2c_1 x(t)\cos\omega_s t + 2c_2 x(t)\cos 2\omega_s t + \cdots$$

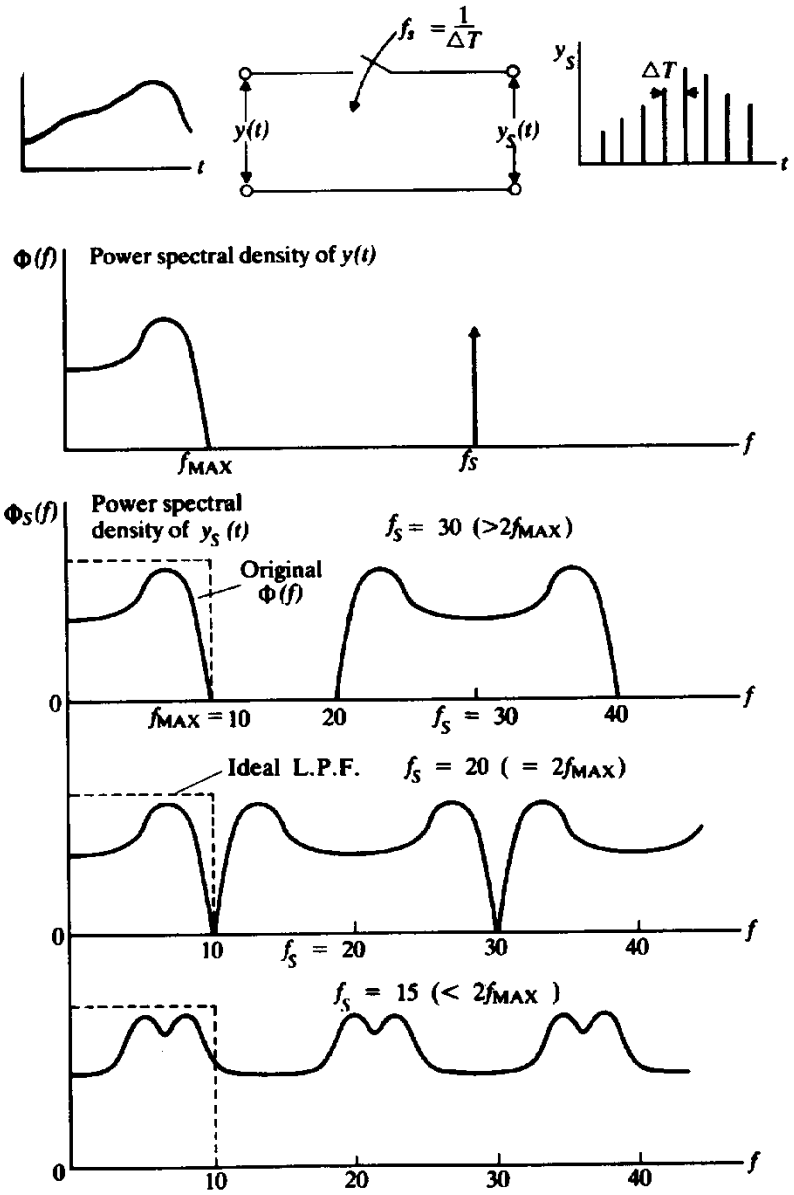
Aliasing

- The Fourier transform of this expression is:

$$X_s = c_0 X(f) + c_1[X(f - f_s) + X(f + f_s)] + c_2[X(f - 2f_s) + X(f + 2f_s)] + \cdots$$

- So the sampled spectrum contains higher frequency copies of the original signal. Aliasing occurs when the copies overlap.

Aliasing



GCU Glasgow Caledonian University

Aliasing - Example

- A signal with real components at 25,70,160 and 510 Hz is sampled at 100 Hz.

- Calculate the set of frequencies observed using the formula:

$$X_s = c_0 X(f) + c_1[X(f - f_s) + X(f + f_s)] + c_2[X(f - 2f_s) + X(f + 2f_s)] + \cdots$$

Aliasing - Example

- The calculated frequencies can be negative since we have used a double sided frequency representation.

- The sign is essentially due to a phase relationship and any frequency whose absolute value lies within the range 0 – f_s will appear on the Fourier spectrum.

Aliasing - Example

aliasing example

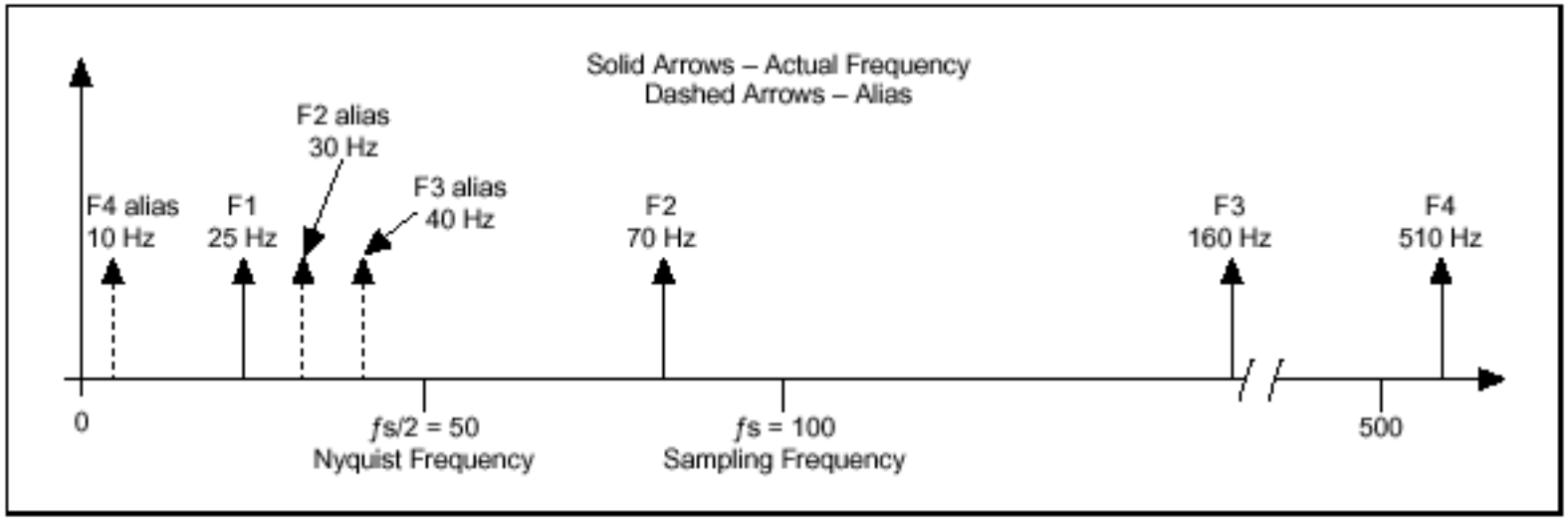
fs= 100

f	f-fs	f+fs	2nd	order	3rd	order	4th	order	5th	order	6th	order
25	-75	125	-175	225	-275	325	-375	425	-475	525	-575	625
70	-30	170	-130	270	-230	370	-330	470	-430	570	-530	670
160	60	260	-40	360	-140	460	-240	560	-340	660	-440	760
510	410	610	310	710	210	810	110	910	10	1010	-90	1110

cells containing $|f \pm n*fs| < fs$

25	75	0	0	0	0	0	0	0	0	0	0	0
70	30	0	0	0	0	0	0	0	0	0	0	0
0	60	0	40	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	10	0	90	0

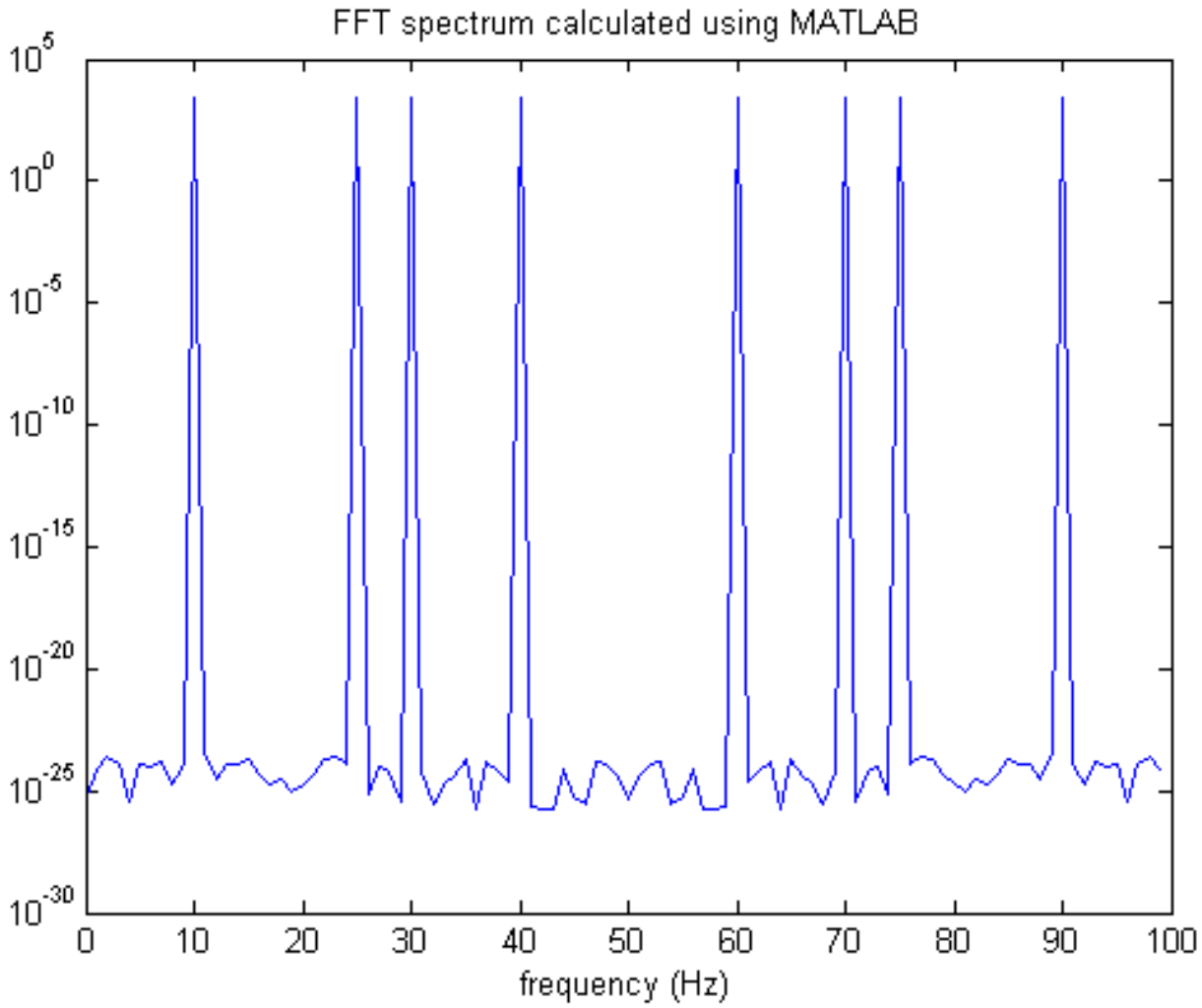
Aliasing - Example



Aliasing - Example

- Notice that the spectrum lines come in pairs: 10 & 90 Hz, 25 & 75 Hz, 30 & 70 Hz, 40 & 60 Hz.
- The whole spectrum presents mirror symmetry.
- However, only the 25 Hz line is real.
- The rest are aliasing artefacts.

Aliasing - Example



More Aliasing!

Analogue to Digital Conversion

- Whenever an analogue signal is to be processed by a digital system it has to be converted to a digital form. The major concerns of converting an analogue signal into digital form are:
  - ensuring that the level is sampled with sufficient resolution; and
  - ensuring that the signal is sampled at a sufficiently high frequency.

Analogue to Digital Conversion

- The sampling theorem has been introduced previously and it implies that the conversion process has to be fast enough to ensure that sampling is carried out at a rate that is at least twice the highest frequency present.

- For the ADC, the conversion time T_{conv} must be less than or equal to $1/f_s$

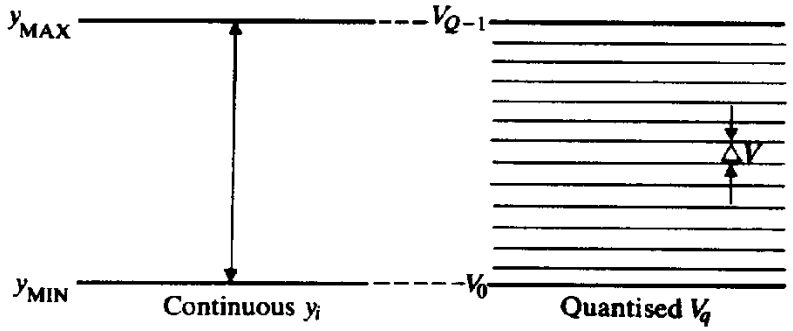
Analogue to Digital Conversion

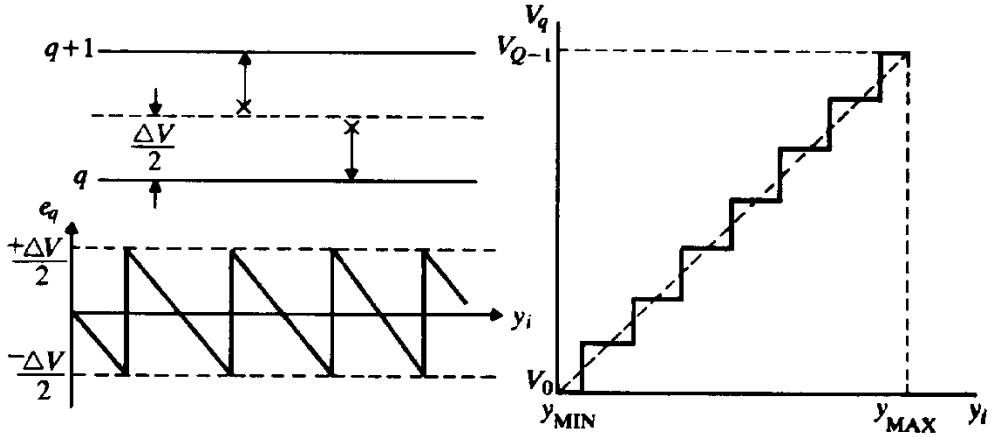
- The resolution of the signal level is determined by the number of binary digits that the signal level is converted to. For example, an '8 bit' converter will convert the analogue signal into one of 256 levels whereas a '10 bit' converter will have 1024 possible levels for signal size. This fact, coupled with a knowledge of the input range of the converter will enable the designer to calculate the resolution of the converter in mV. This is summarised on the next slide for a converter with an input range of 5V.

ADC Resolution

n-bit ADC	Number of channels	Channel resolution (mV)
8	256	19.5
10	1024	4.89
12	4096	1.22
16	65535	0.076

Quantisation





Quantisation

- N-bit A/D converter splits input range into 2^n-1 divisions
- 2^n output values from 0 $\rightarrow 2^n$-1

- Resolution $$\frac{1}{2^n - 1} \times 100\%$$

- Quantisation error $$\frac{0.5}{2^n - 1} \times 100\%$$

Analogue to Digital Conversion

- The resolution in terms of mV is not the most useful fact and one is usually more concerned with the resolution in % terms of the quantity being measured.
- The upper limit of resolution can be obtained by mapping the range of the measurand (not the transducer) to the A/D converter input range.
- This process may involve amplification/ attenuation and d.c shifting.

Analogue to Digital Conversion

- Assuming optimum range matching, one is able to calculate the minimum number of bits required for the A/D converter to give a particular resolution in the measurand.
- However, it should be remembered that the calculation gives the minimum number of bits required and it may be useful to build in some factor of safety by mapping a larger measurand range to the A/D converter range.

Analogue to Digital Convertors

- The operation of the majority of A/D converters is based on the principle of null measurement.
- In this case a reference voltage is scaled in accordance with a digital code and the resultant voltage is compared with the unknown input voltage using a high gain differential amplifier (comparator).
- The code is adjusted to obtain zero output from the difference detector (null condition) and the setting is used as a measure of input voltage.
- A device that scales a reference voltage in accordance with a digital input is termed a digital to analogue converter.

Analogue to Digital Convertors

- The means of arriving at the null point distinguishes two types of converter, namely the ramp and the successive approximation converter.
- In the ramp converter the scaled reference voltage is increased in a linear fashion from zero until the null point is reached. At this point the process is terminated and the input to the scaler (digital to analogue converter) is used as the measure of input voltage. Although the control of the scaler is simple in that it is fed with an upwards counting digital sequence the conversion time is variable.

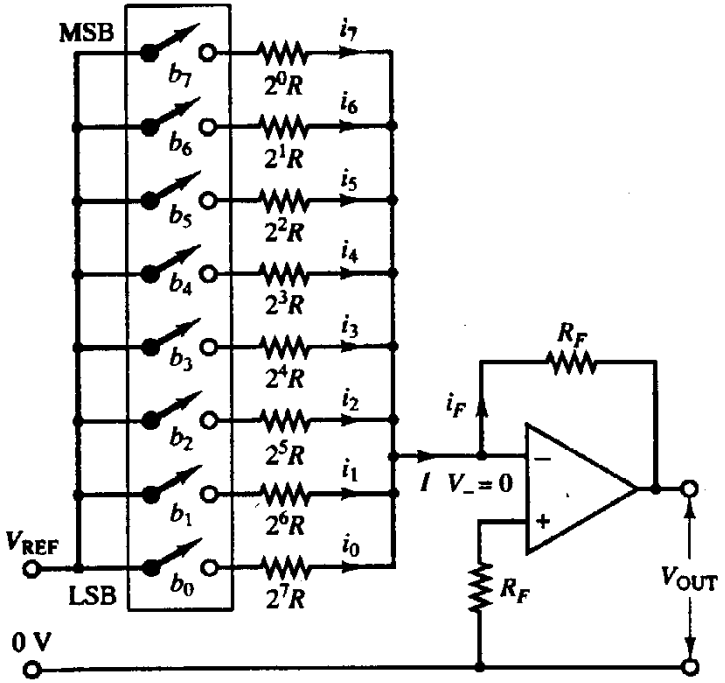
Analogue to Digital Convertors

- The successive approximation converter operates by starting with the most significant bit to the scaler.
- Once this bit is set the polarity of the imbalance is used to indicated whether n the second cycle (when the second msb is to be set regardless) the msb should be left set or whether it should be reset.
- This cycle is then repeated for all bits of the converter. Accordingly the conversion time does not depend on the input signal but is a function of the number of bits and the clock speed.

Analogue to Digital Convertors

- Most ADC's work by comparing input voltage with an internally generated reference voltage.

- For this we need a D/A converter.



$b_i = 1$ – switch closed, $b_i = 0$ – switch open

$i_F = I = i_7 + i_6 + i_5 \ldots + i_1 + i_0$

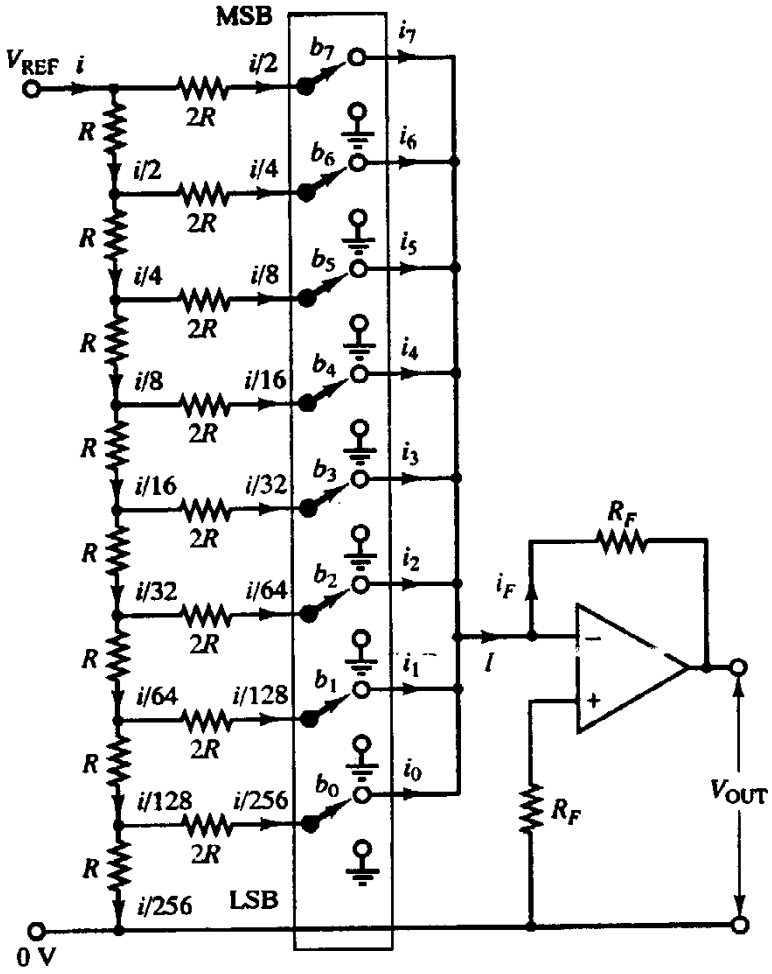
but $i_F = \dfrac{0 - V_{OUT}}{R_F}$

i.e. $V_{OUT} = -R_F(i_7 + i_6 + i_5 + \ldots + i_1 + i_0)$

$i_7 = \dfrac{V_{REF}}{2^0R}\, b_7,\ i_6 = \dfrac{V_{REF}}{2^1R}\, b_6, \ldots, i_1 = \dfrac{V_{REF}}{2^6R}\, b_1,\ i_0 = \dfrac{V_{REF}}{2^7R}\, b_0$

$$V_{OUT} = \frac{-R_F}{R} V_{REF} \left[\frac{b_7}{2^0} + \frac{b_6}{2^1} + \frac{b_5}{2^2} + \ldots \frac{b_1}{2^6} + \frac{b_0}{2^7}\right]$$

(a)



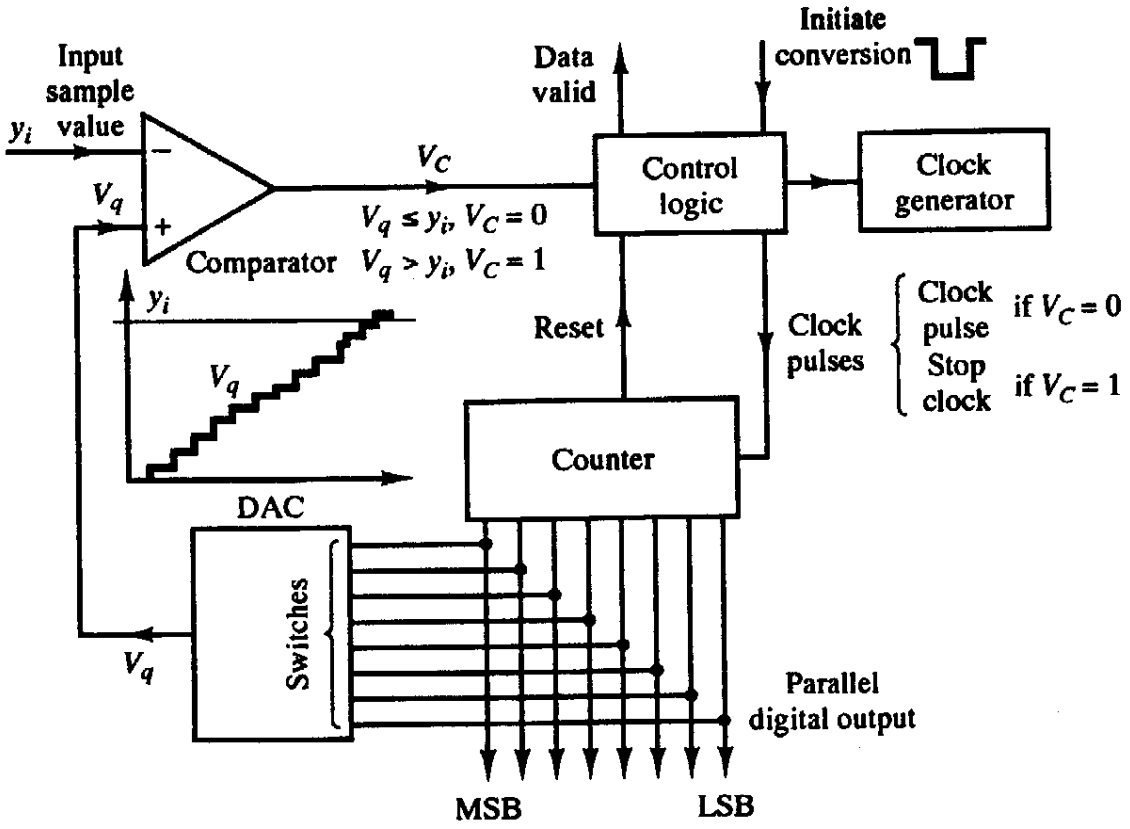
$b_i = 1$ – switch up $b_i = 0$ – switch down

$i_7 = i/2\ b_7,\ i_6 = i/4\ b_6,\ i_5 = i/8\ b_5, \ldots i_1 = i/128\ b_1,\ i_0 = i/256\ b_0$

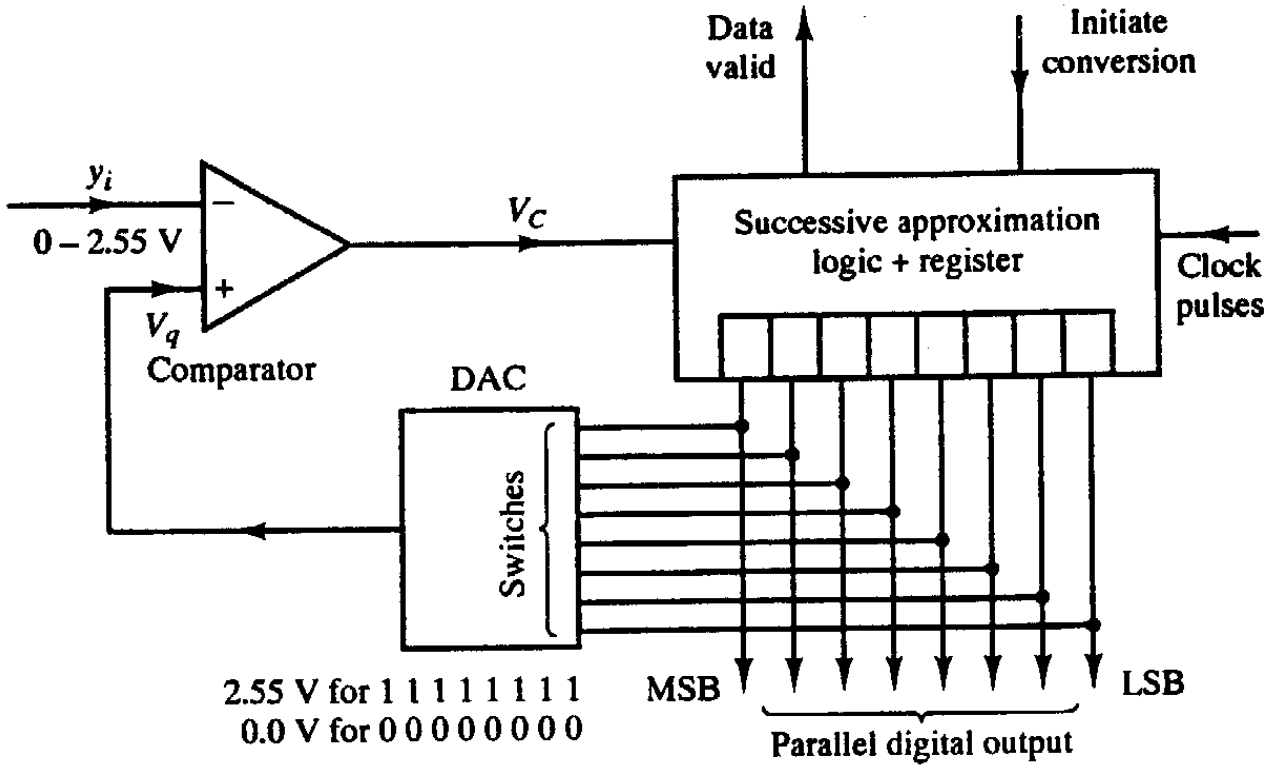
$$V_{OUT} = \frac{-R_F i}{2} \left[\frac{b_7}{2^0} + \frac{b_6}{2^1} + \frac{b_5}{2^2} + \ldots \frac{b_1}{2^6} + \frac{b_0}{2^7}\right]$$

(b)

Ramp ADC



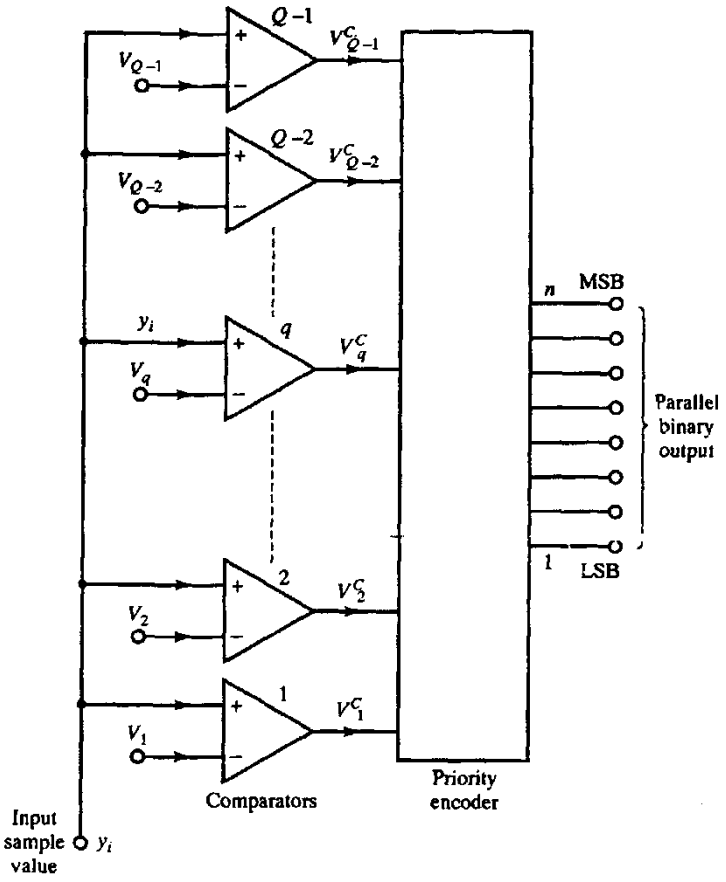
Successive Approximation ADC



Successive Approximation Calculation - Example

	Input voltage $y_i = 0.515$ V				
	Clock pulse	DAC input	DAC V_q output volts	Comparator output V_C	Result
Initiate conversion →	1 Clear register	00000000	0	0	
	2 First guess	01111111 $(127)_{10}$	1.27	1 HIGH	$b_7 = 0$
	3 Next guess	00111111 $(63)_{10}$	0.63	1 HIGH	$b_6 = 0$
	4	00011111 $(31)_{10}$	0.31	0 LOW	$b_5 = 1$
	5	00101111 $(47)_{10}$	0.47	0 LOW	$b_4 = 1$
	6	00110111 $(55)_{10}$	0.55	1 HIGH	$b_3 = 0$
	7	00110011 $(51)_{10}$	0.51	0 LOW	$b_2 = 1$
	8	00110101 $(53)_{10}$	0.53	1 HIGH	$b_1 = 0$
	9 Final guess	00110100 $(52)_{10}$	0.52	1 HIGH	$b_0 = 0$
Data valid →	Output digital signal = 00110100				

Flash ADC



Time Division Multiplexing