

A stylized graphic of a vortex or swirl, rendered in shades of blue and white, with the word "Anadigmvortex" overlaid in a bold, italicized blue font.

Anadigmvortex

Sensor signal conditioning with programmable analog ICs



© Anadigm 2003. All rights reserved.

- **Programmable Analog ICs**
- **The AnadigmVortex FPAA**
- **EDA tools**
- **Sensor interfacing**



Programmable Analog ICs

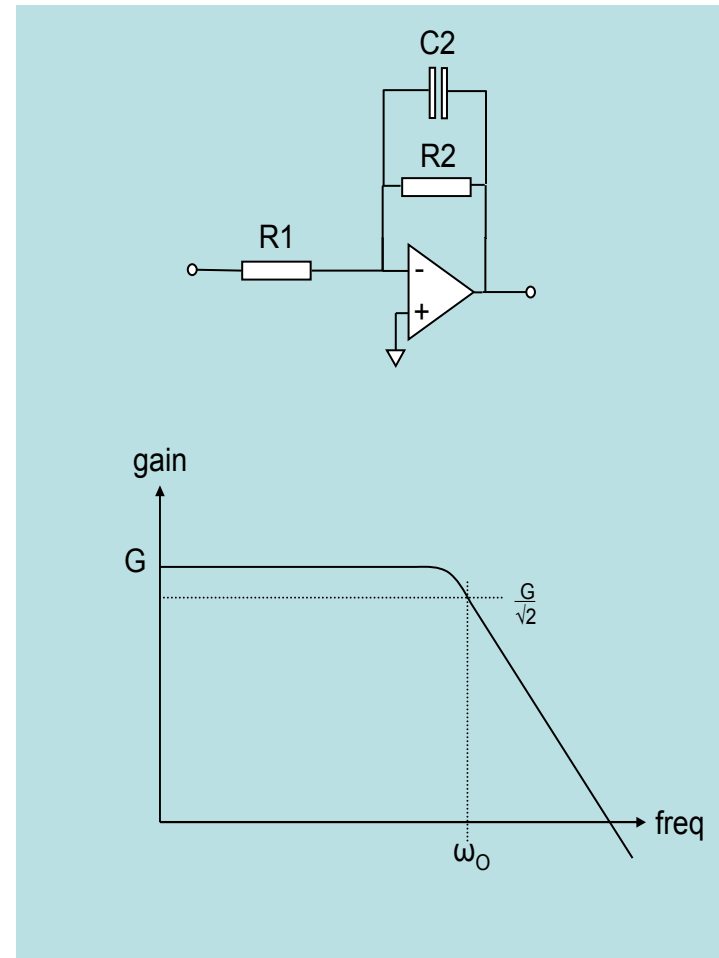
Example analogue circuit:

- **Active low-pass filter**

$$G = -\frac{R2}{R1}$$

$$\omega_0 = \frac{1}{R2.C2}$$

- Frequency parameter dependent on *absolute* resistance and capacitance values: sensitive to process variations, ageing and temperature



Achieving high-accuracy

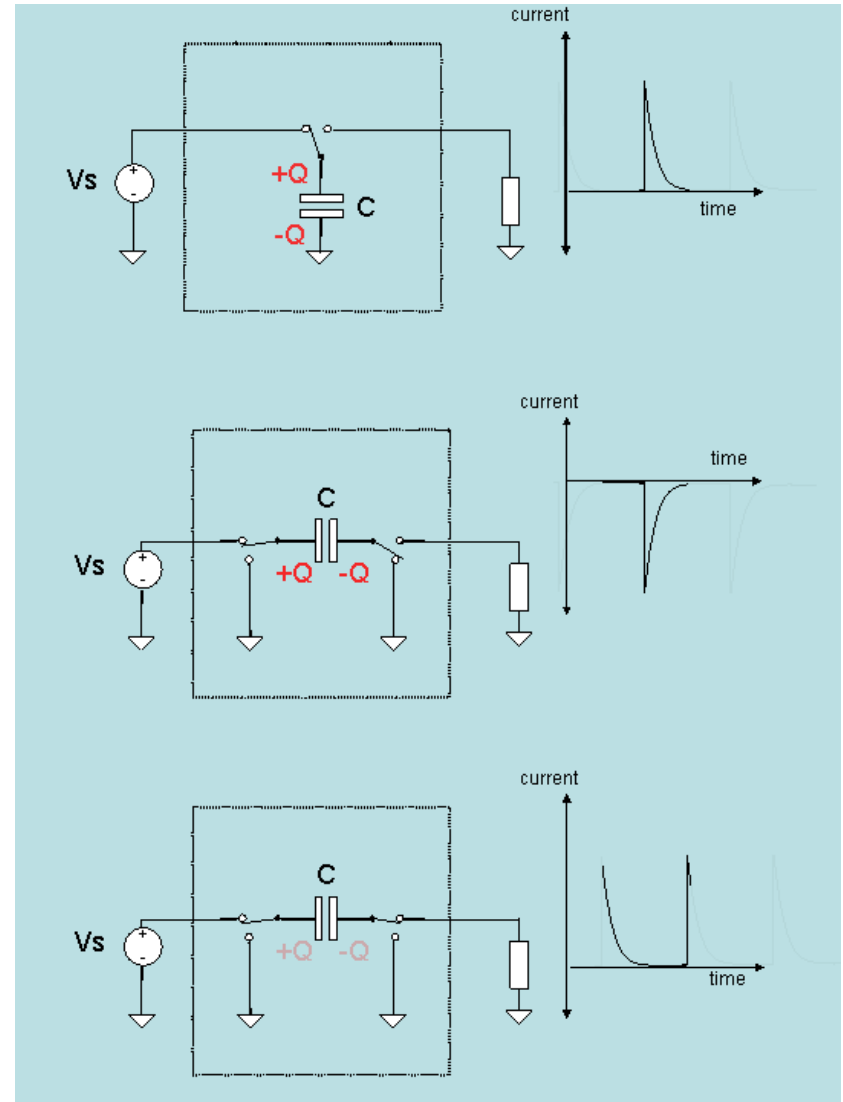
- **Use external (discrete) components (but watch temperature dependence)**
- **Use ratio-dependent building blocks**
- **Use SWITCHED CAPACITOR TECHNIQUES**
 - SC circuits can provide the advantages of digital circuits in the analogue world
 - programmability and accuracy

What is a switched-capacitor?

- **Charge-per-clock-cycle**
= $V_s \cdot C$
- **Average current**
= $V_s \cdot C \cdot f_c$
- **Equivalent resistance**

$$R = \begin{cases} + \frac{1}{C \cdot f_c} \\ \text{or} \\ - \frac{1}{C \cdot f_c} \end{cases}$$

provided $f_c \gg f_s$



Why Switched Capacitor?

- **Much larger resistance values for a given area**
- **Better “resistor” values**
 - Better tolerance; typically $\pm 1.0\%$
 - Better matching - typically $\pm 0.1\%$
 - Better temperature coefficients - $\sim 1/20\text{th}$ to $1/300\text{th}$
 - Better voltage linearity - $\sim 1/5\text{th}$ to $1/2000\text{th}$
 - Wider range
- **Temperature and process independent ratios**
- **The “extras”**
 - Building blocks with “Negative Resistors”
 - Non-linear functions (rectification)
 - Corner frequencies scale linearly with the sample clock

Switched-capacitor functions (1)

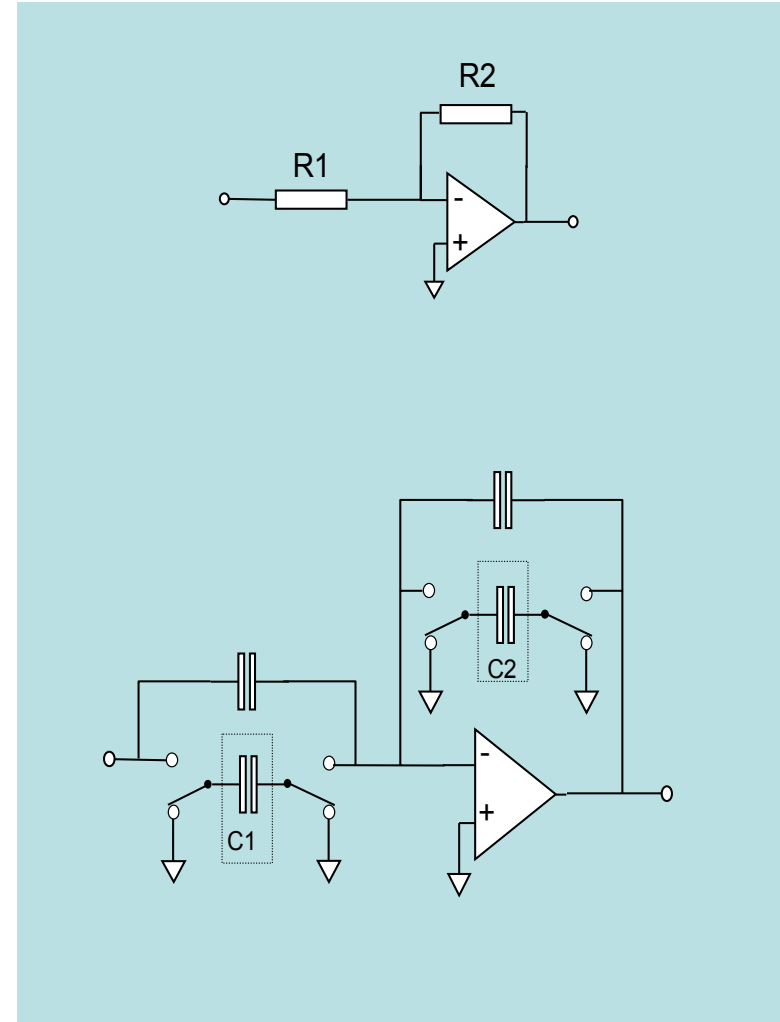
- **Inverting gain-stage**

- Gain defined by *capacitor ratio*

$$G = -\frac{R2}{R1} = -\frac{1/C2 \cdot f_c}{1/C1 \cdot f_c} = -\frac{C1}{C2}$$

$$V_{out} = V_{in} \cdot G$$

NB: Non-switched capacitors prevent opamp being left “open-loop”



Switched-capacitor functions (2)

- **Non-inverting gain-stage (& LPF)**

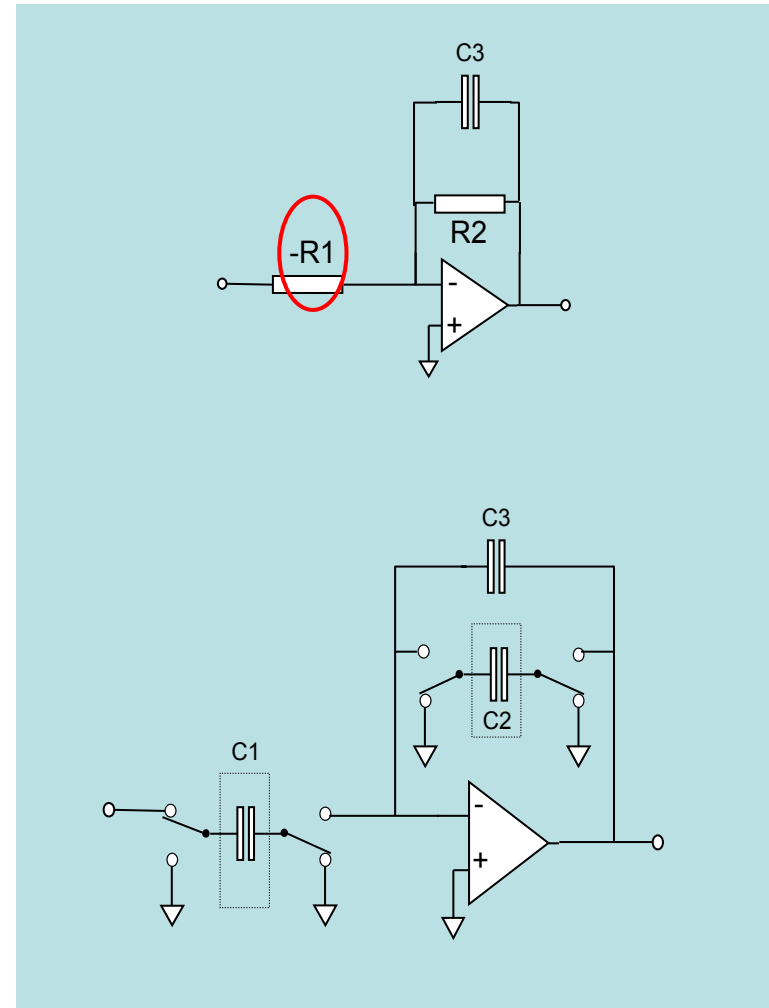
- change input sampling phase

$$G = + \frac{C1}{C2}$$

NB: Non-switched capacitor needed to prevent opamp being left “open-loop”

- creates low-pass filter

$$\omega_o = \frac{1}{R2.C3} = \frac{f_c.C2}{C3}$$



Switched-capacitor functions (3)

- **Summing gain-stage (& LPF)**

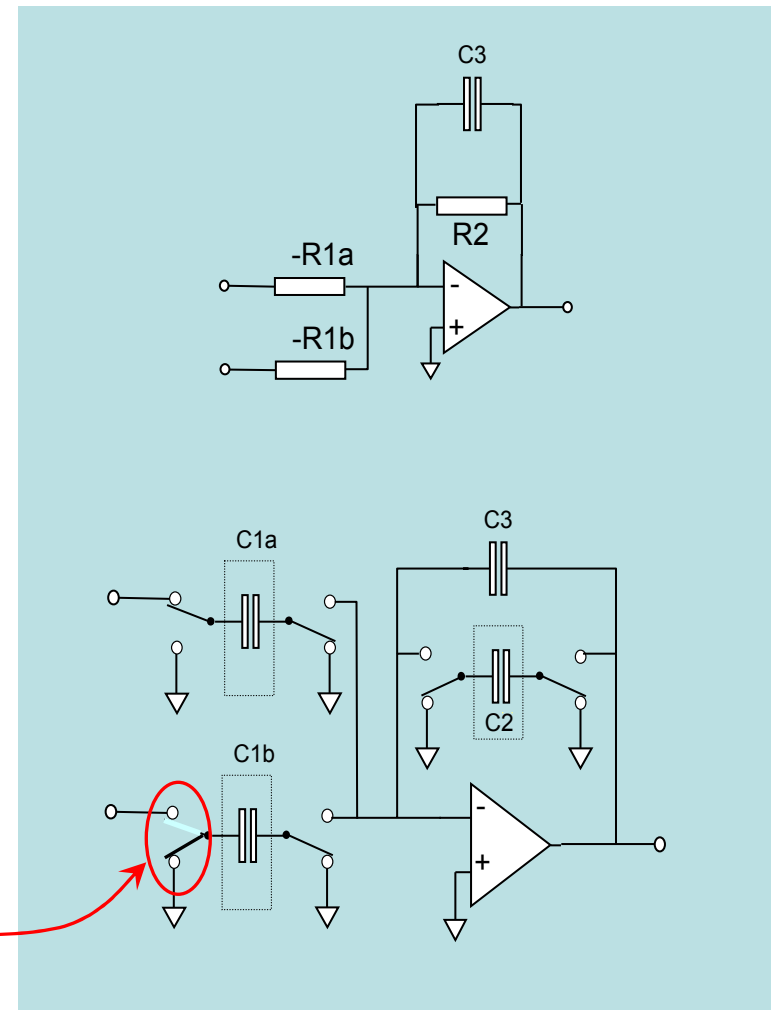
$$G1 = + \frac{C1a}{C2}$$

$$G2 = + \frac{C1b}{C2}$$

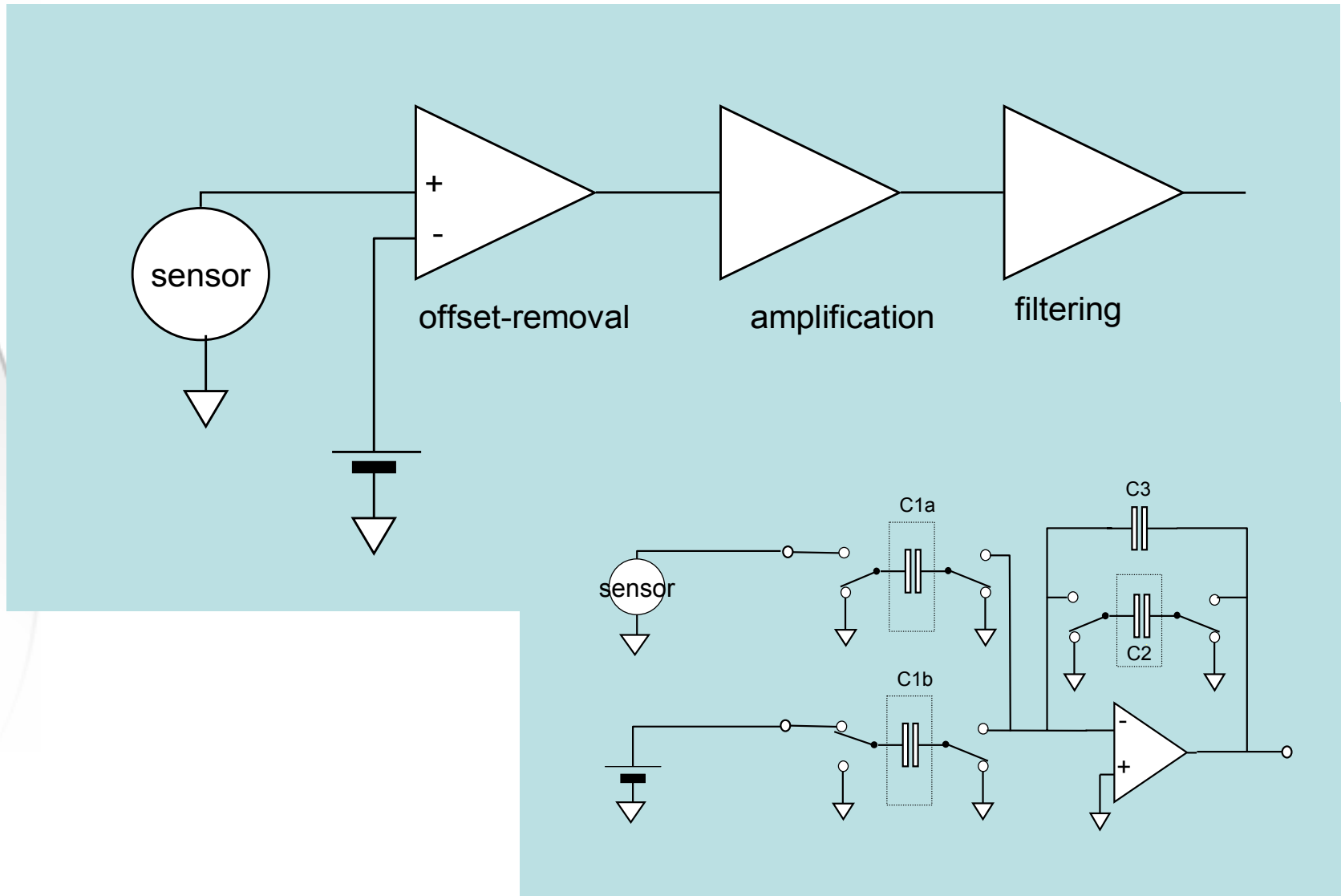
$$V_{out} = V_{in1} \cdot G1 + V_{in2} \cdot G2$$

Reverse phase to get :

$$V_{out} = V_{in1} \cdot G1 - V_{in2} \cdot G2$$



Example analogue signal processing



Switched-capacitor functions (4)

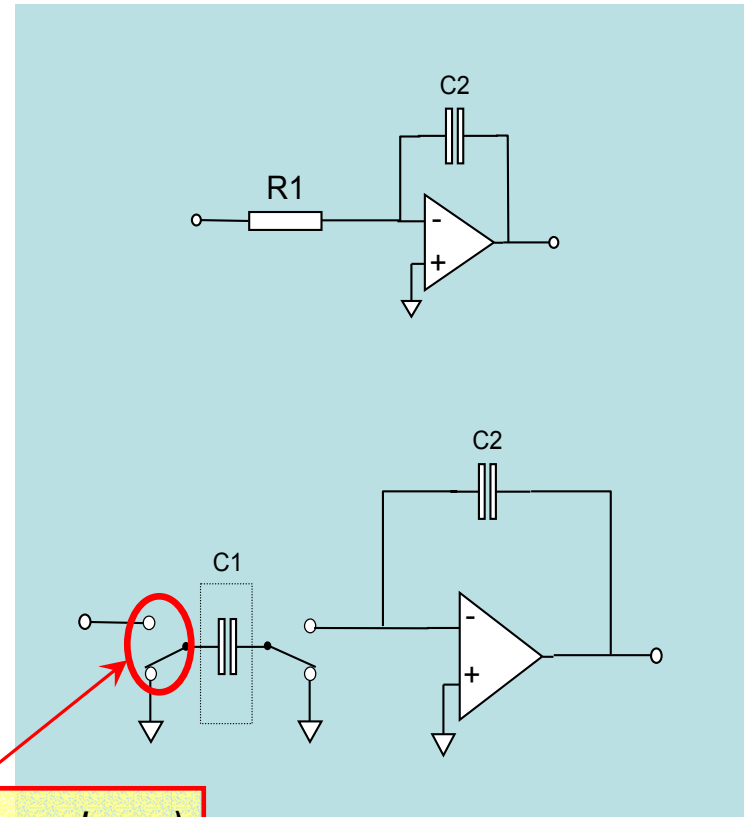
- **Integration**
(common component in control systems)

$$V_{out} = K \int V_{in} . dt$$

$$K = - \frac{1}{R1 . C2}$$

$$= \pm \frac{f_c . C1}{C2}$$

(Sign dependent on input-sampling phase)



Switched-capacitor functions (5)

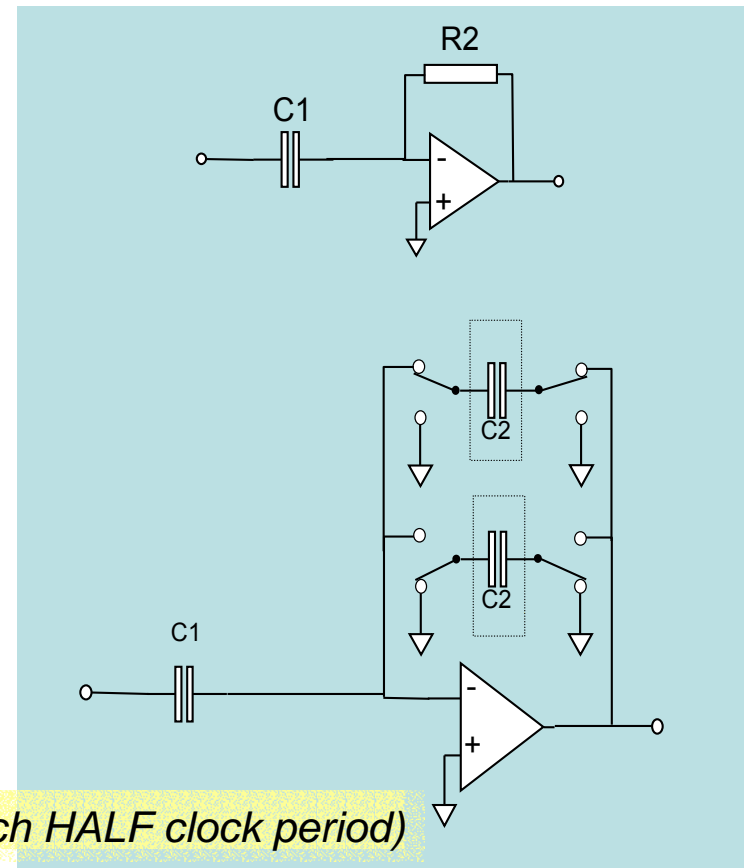
- **Differentiation**
(common component in control systems)

$$V_{out} = K \frac{dV_{in}}{dt}$$

$$K = -C1.R2$$

$$= -\frac{1}{2} \frac{C1}{f_c.C2}$$

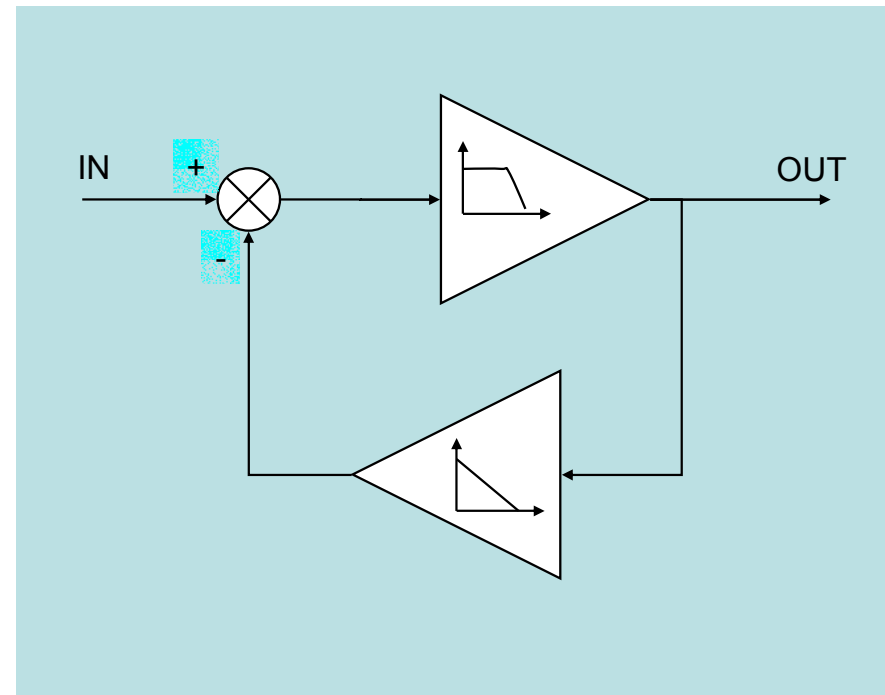
(output responds to input change during each HALF clock period)



Switched-capacitor Filters

- **Biquad stage**

- Replace RLC prototype with signal-flow path equivalent
- Implement using integrator building blocks

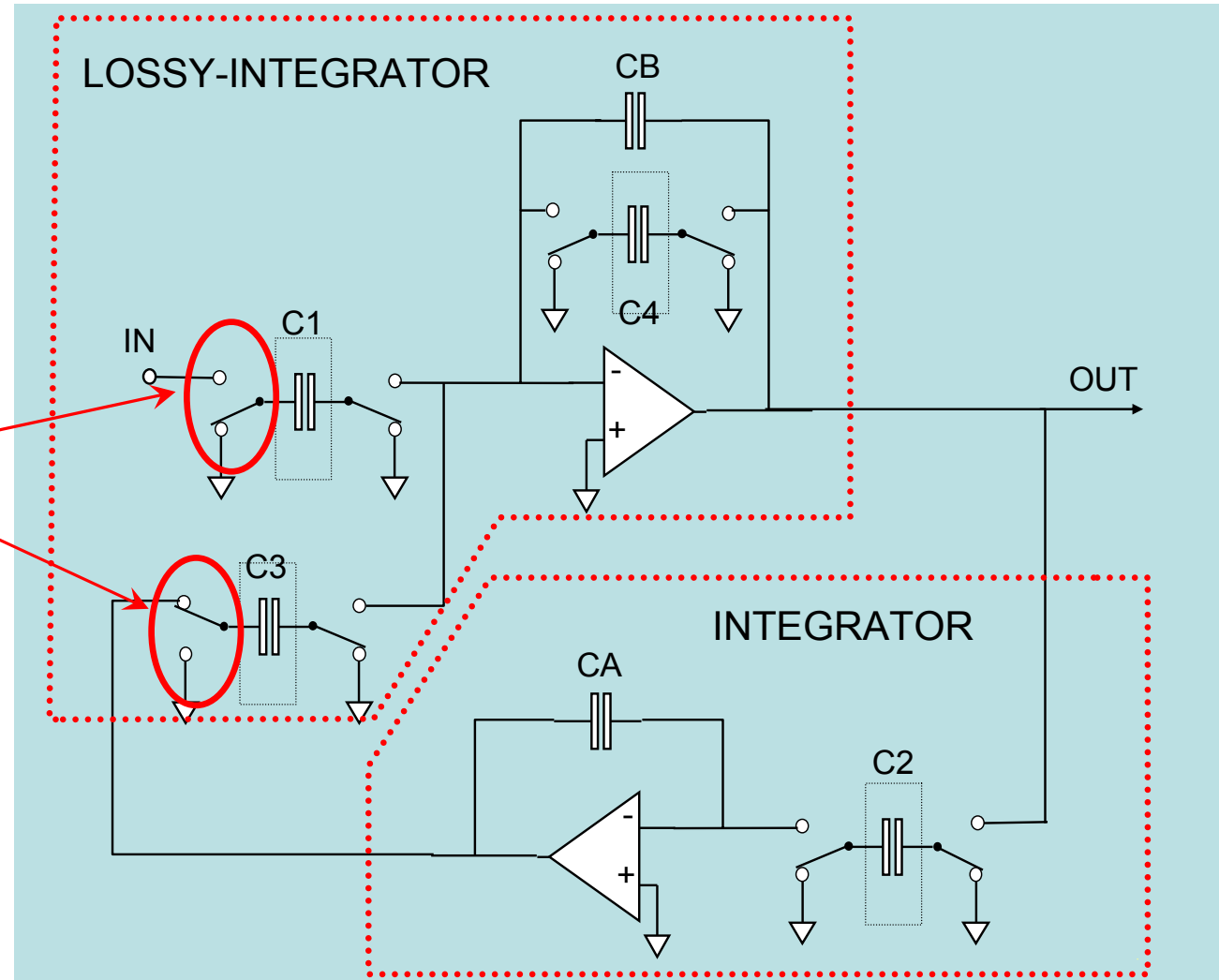


SC Bandpass Filter

$$Gain = -\frac{C1}{C4}$$

$$\omega_o = f_c \sqrt{\frac{C2.C3}{CA.CB}}$$

Different sampling phases

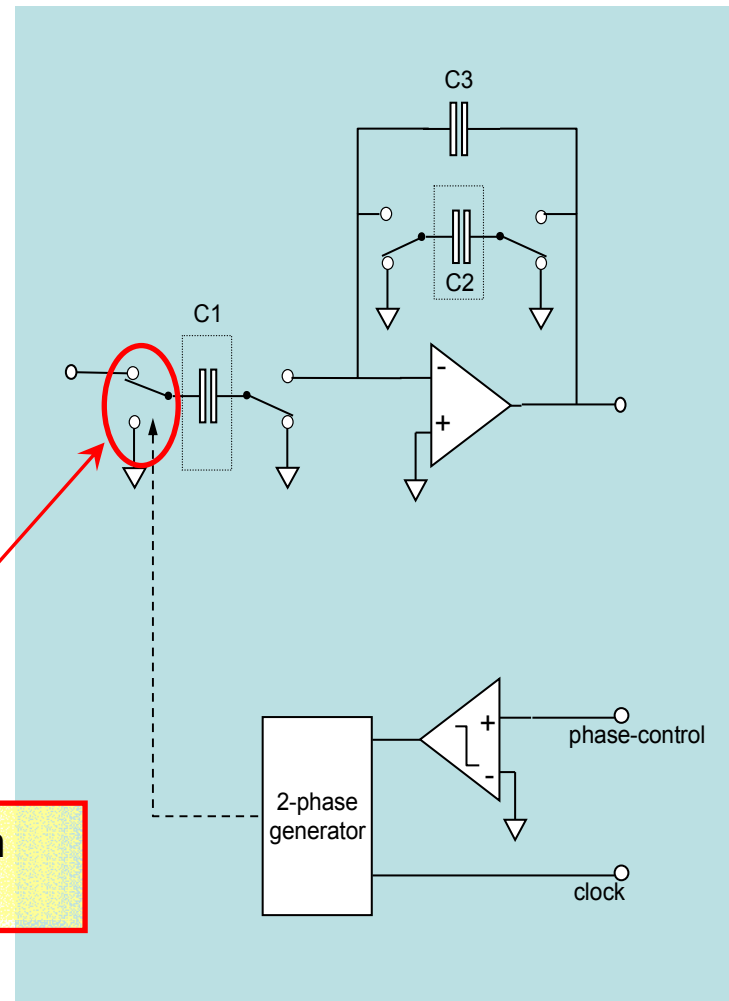


Dynamic Phase-Switching

- **Addition of a comparator and some logic gives increased flexibility**

- e.g. polarity-swapping amplifier

Input sampling-phase dependent on control signal

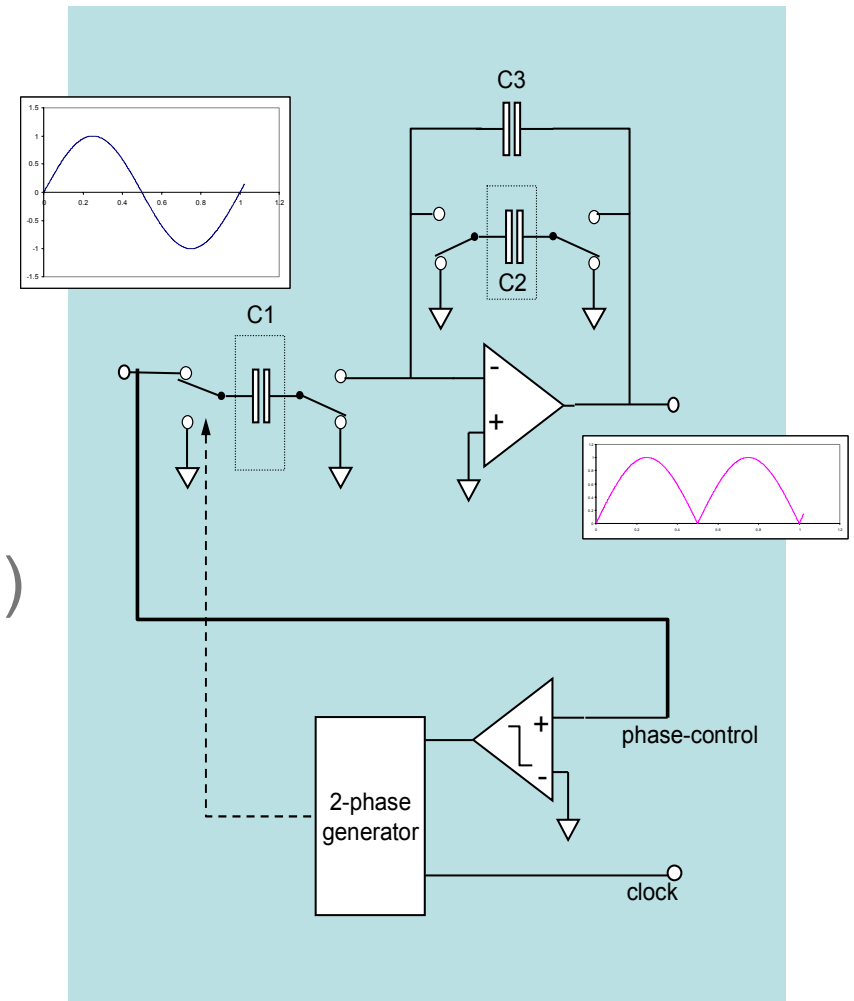


SC rectifiers

- **Full-wave**

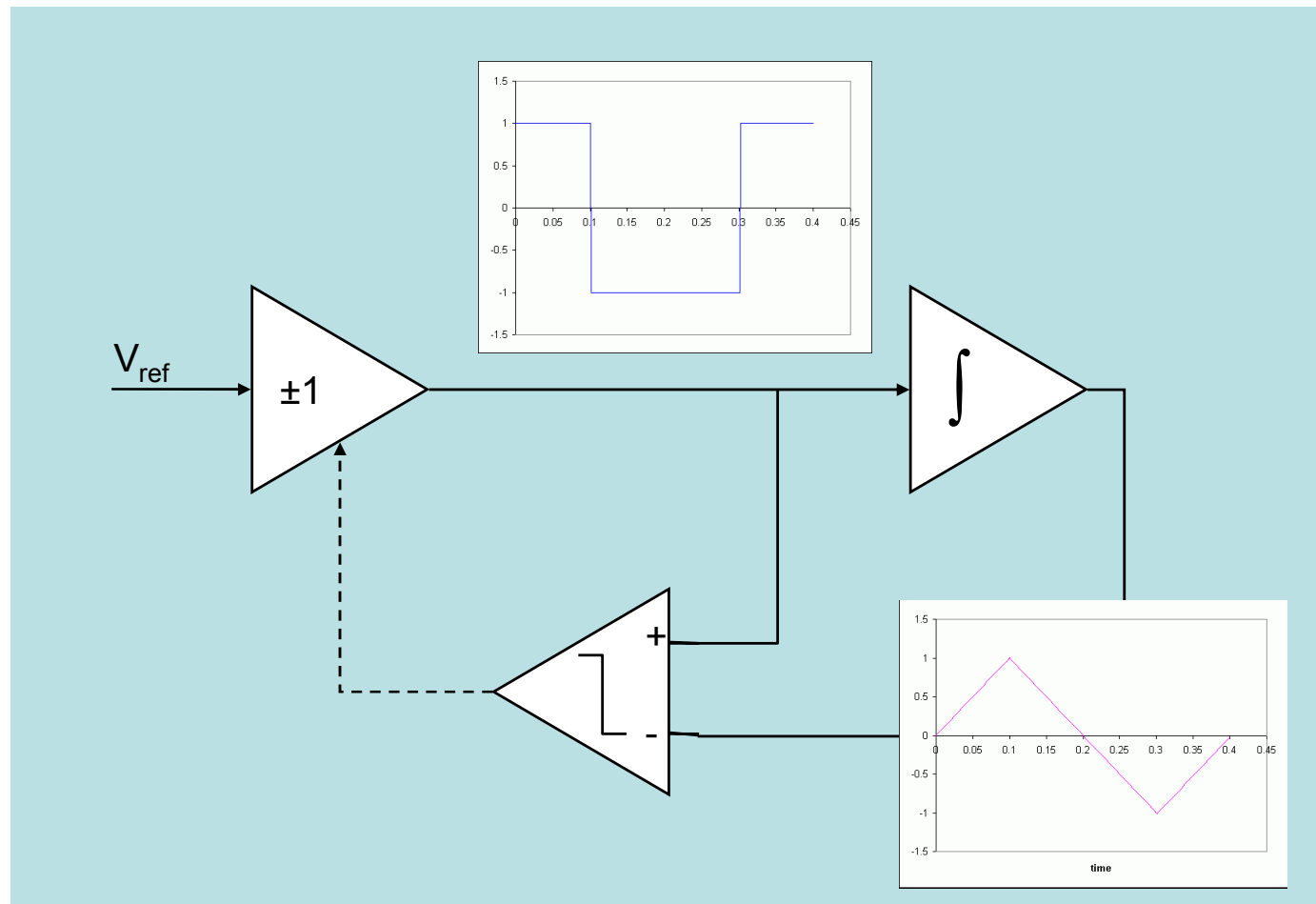
- Use input signal to control amplifier polarity
- Output is always positive (or negative)

- NO DIODES!



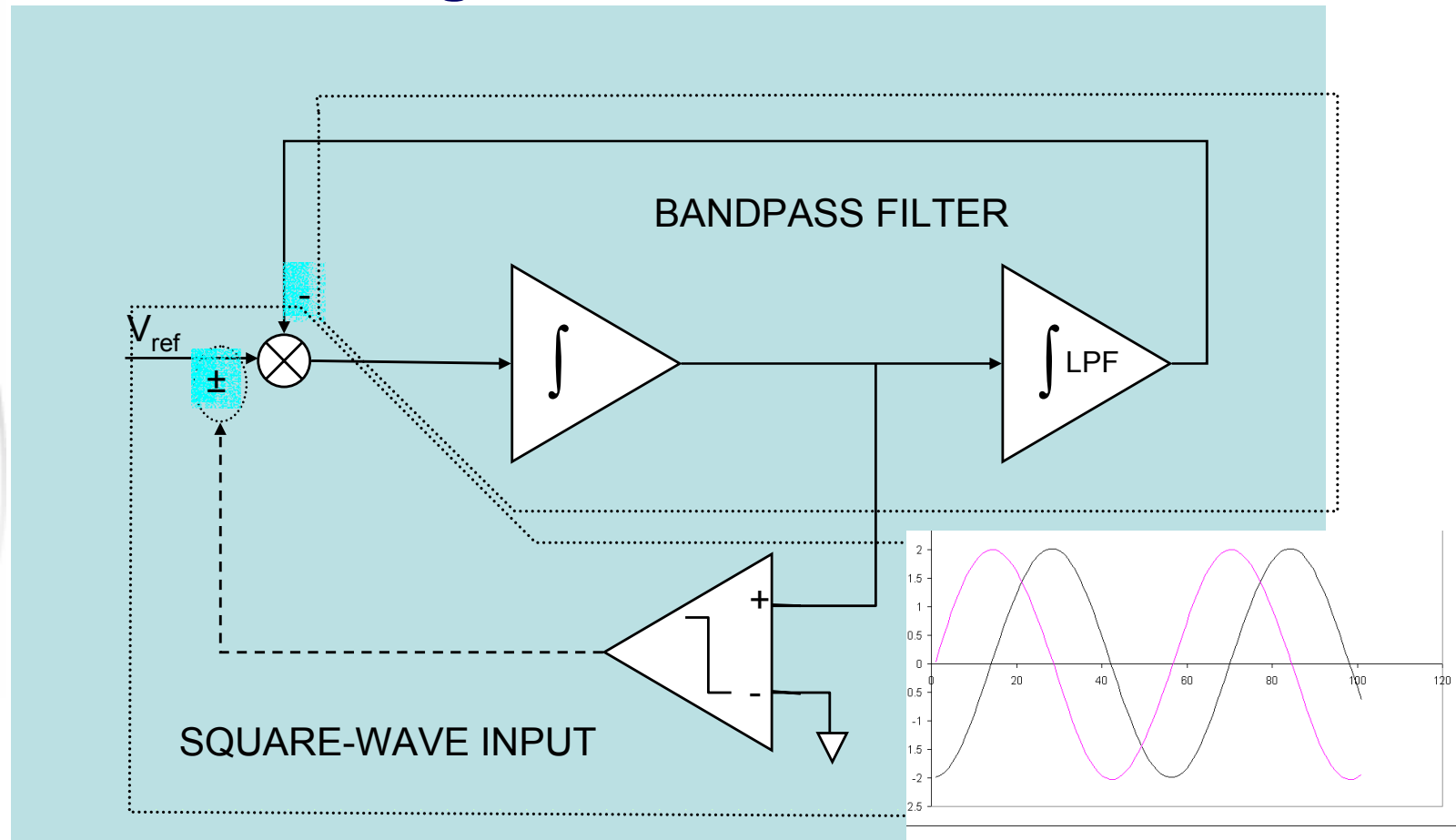
SC oscillators (1)

- Relaxation (ramp generator)



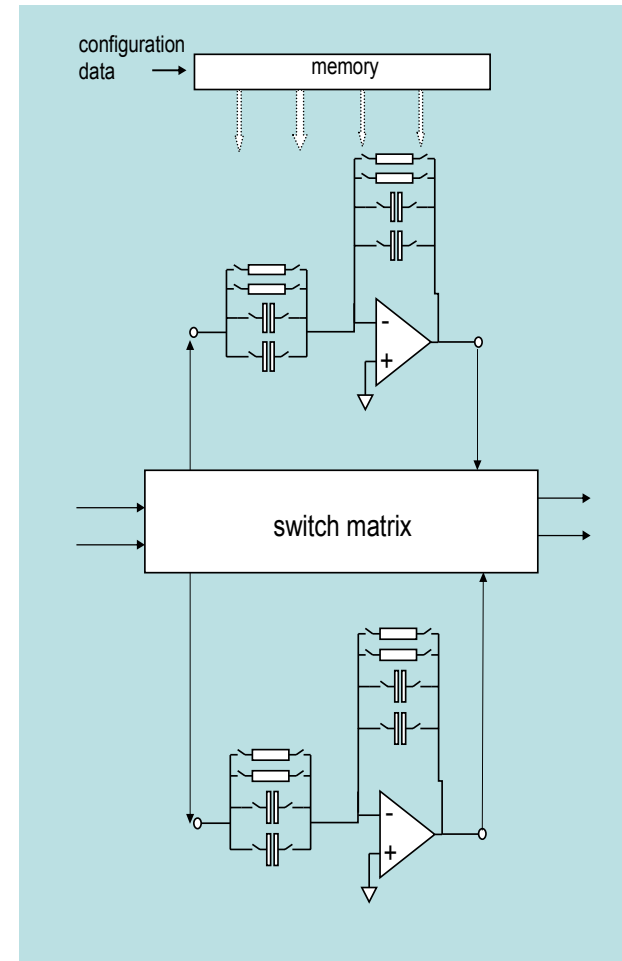
SC oscillators (2)

- **Sine-wave generator**



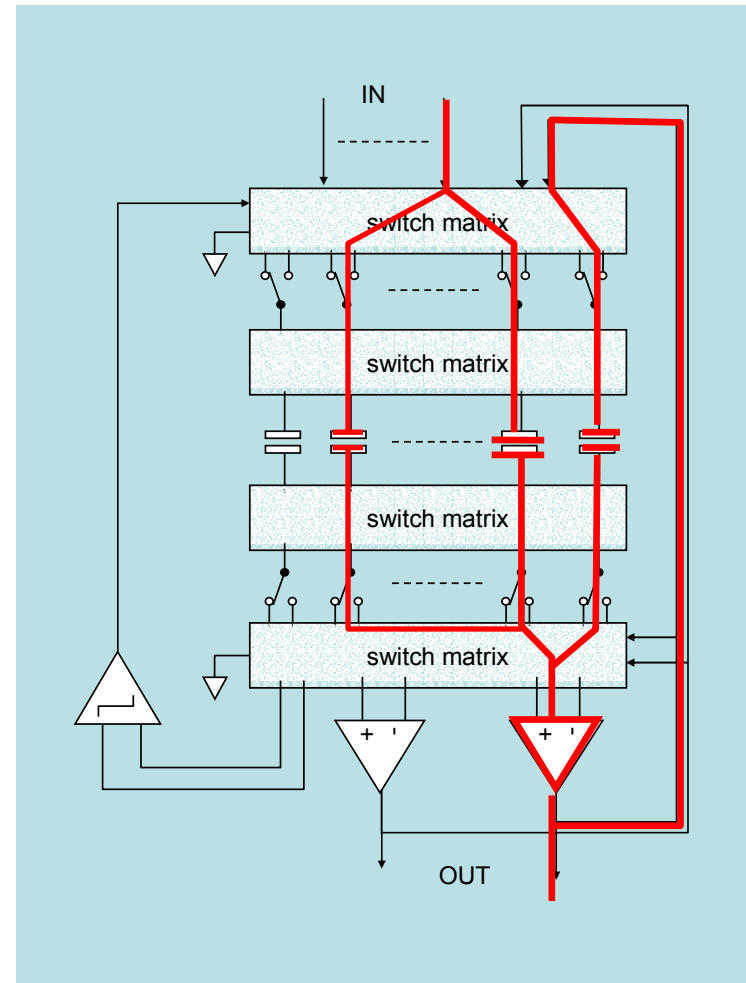
Building programmable analogue circuits

- **Fixed function, variable parameters**
 - Exploit current-dependence of some parameters
- **Variable functionality, variable parameters**
 - Opamp-based topology changed using switches (or anti-fuses)
 - Parameters changed by adding/removing components



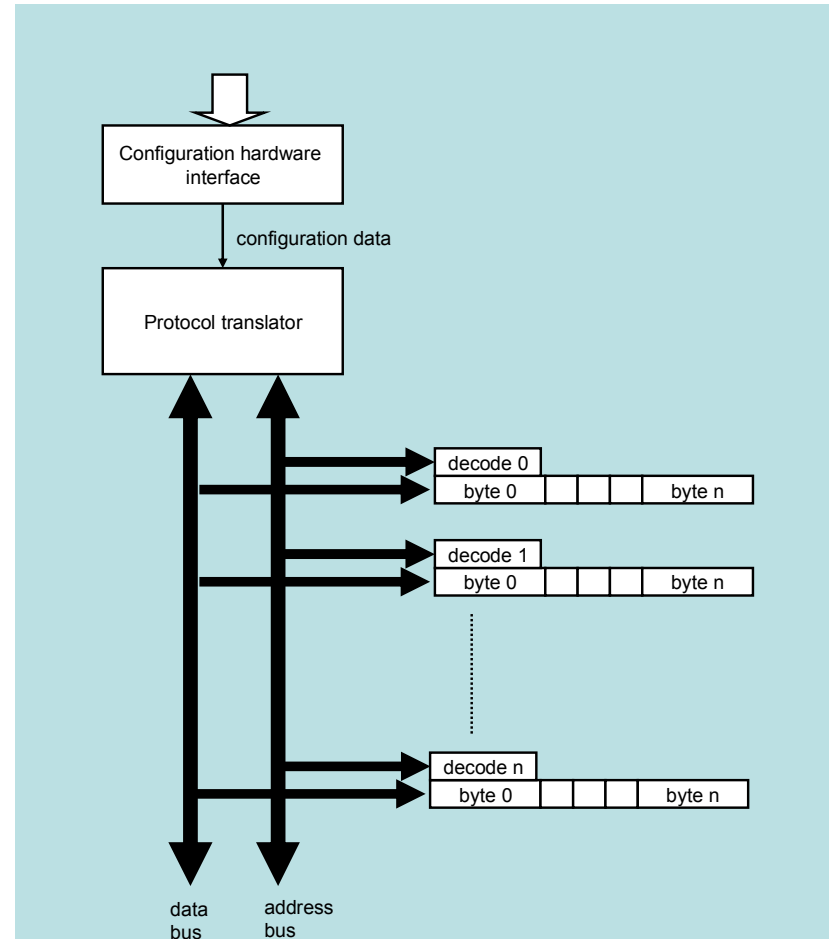
Programmable SC architecture

- **SC techniques can cover many analogue functions:**
 - Gain
 - Sum/Difference
 - Filtering
 - Rectification
 - Oscillators
- **Function parameters AND functionality easily programmed**

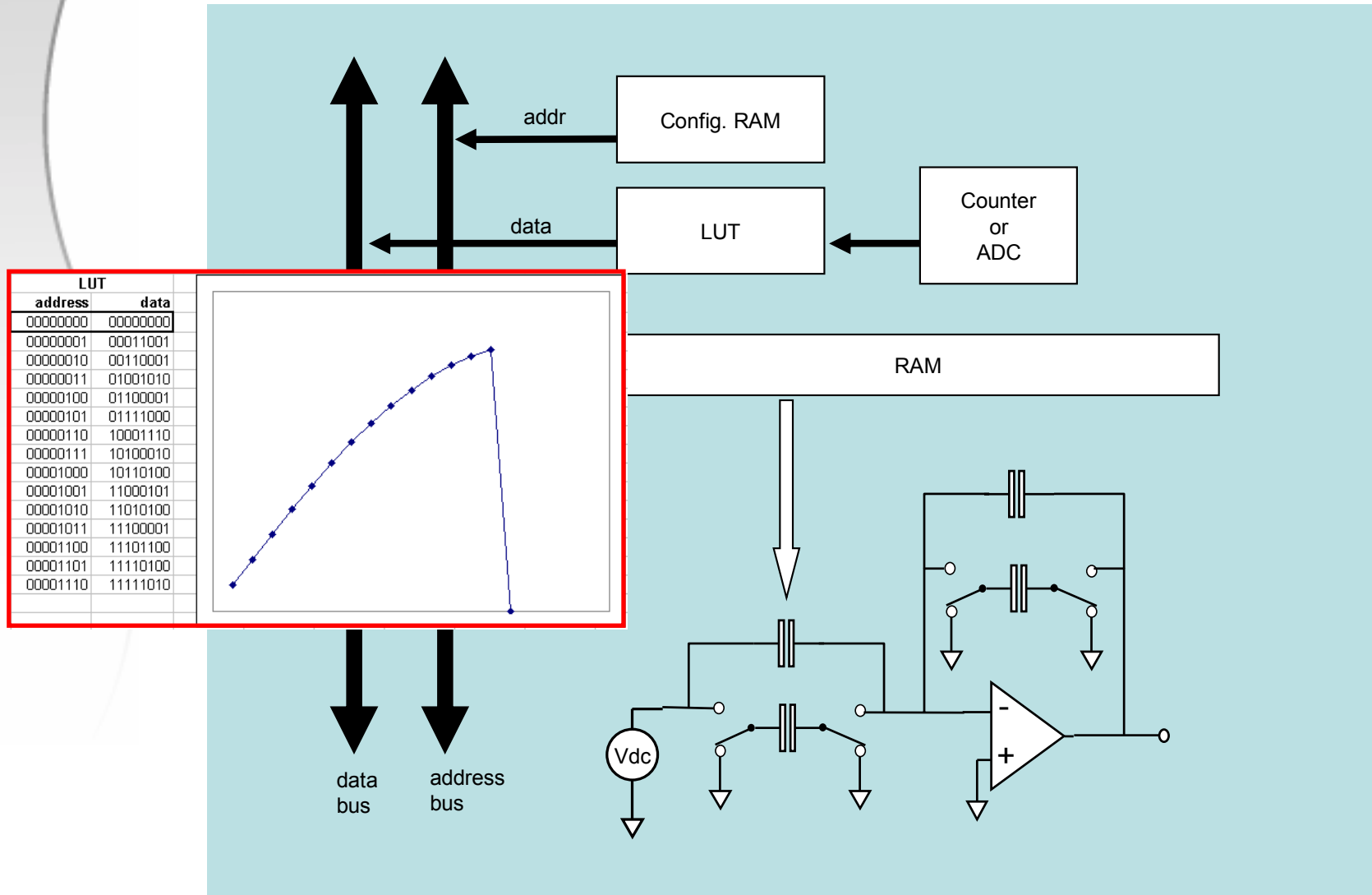


Configuration

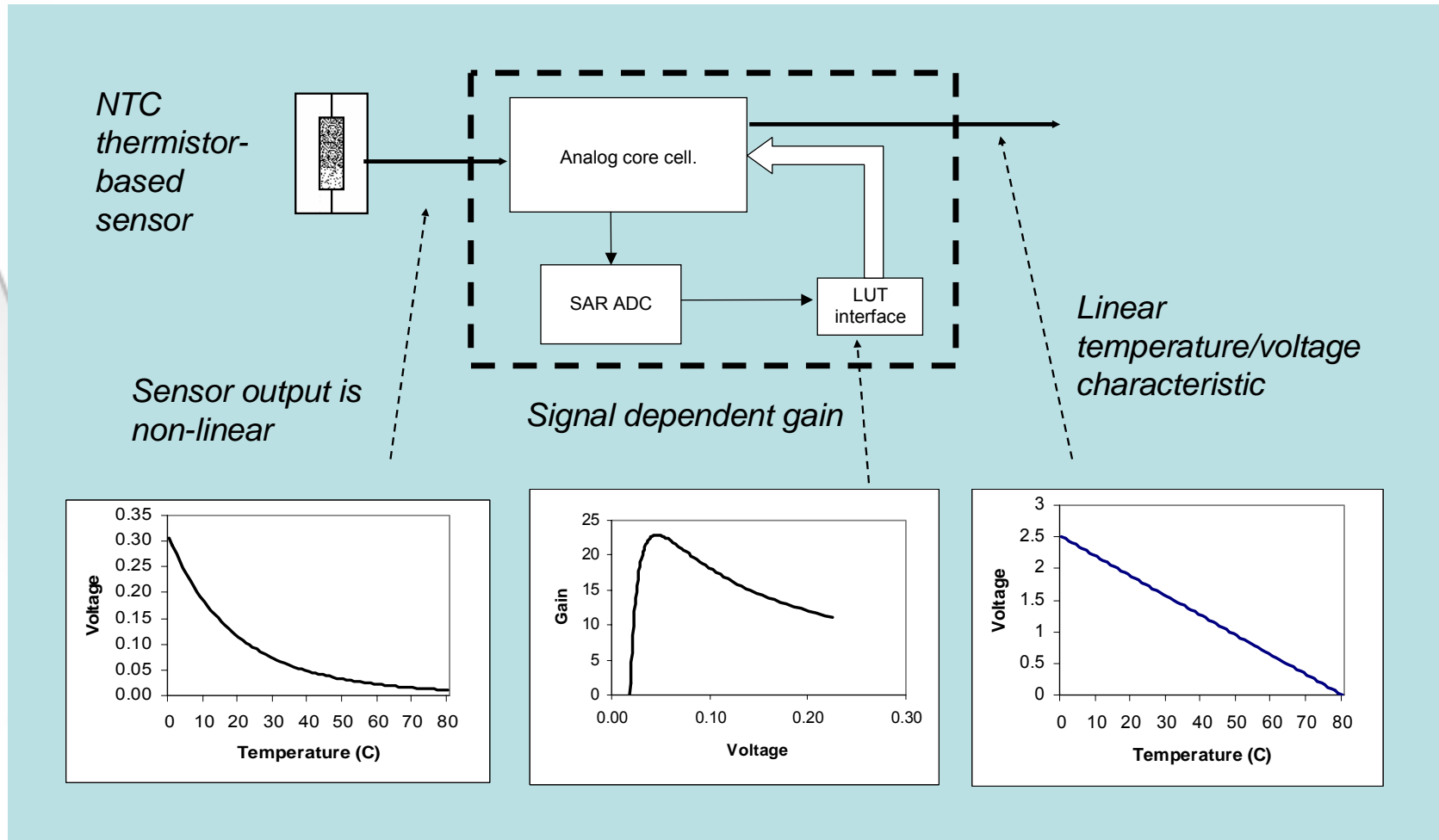
- **Addressable blocks give flexible and fast re configuration**
- **Configuration data decoded to yield programming data/addresses**
- **“Shadow RAM” allows full re configuration in one cycle**



Internal control buses

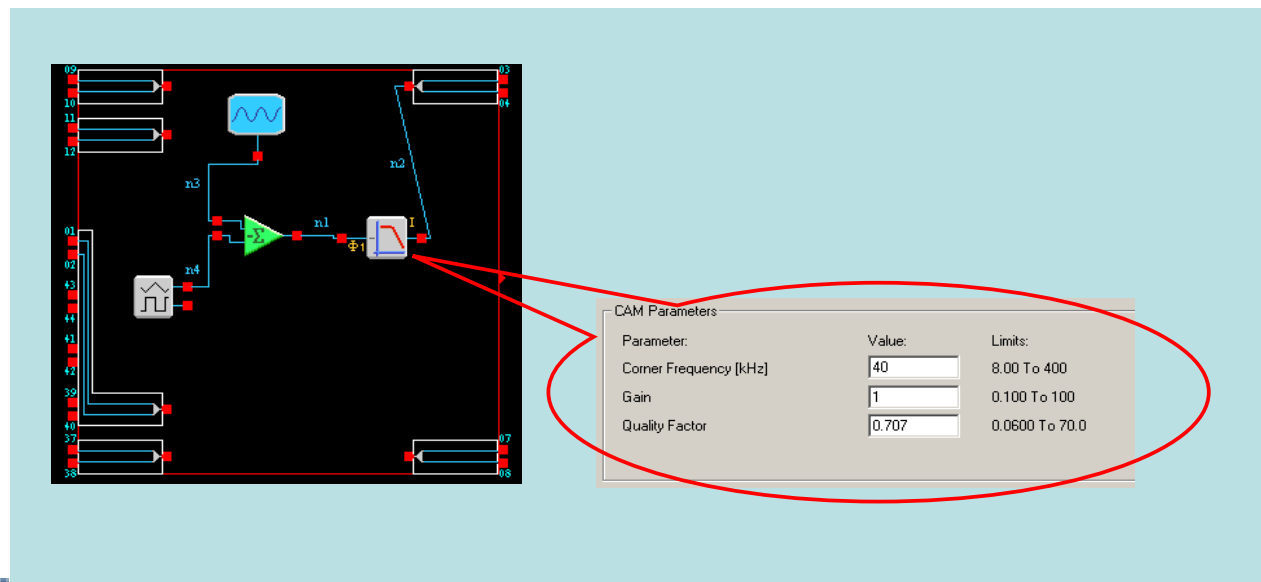


SC linearisation



Summary (1)

- **SC techniques can provide:**
 - High accuracy integrated components
 - Process, temperature, & supply insensitivity
 - “Correct-by-construction” analogue functions
 - Design at a high-level of abstraction



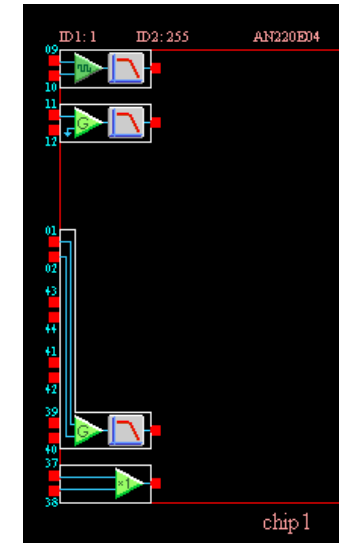
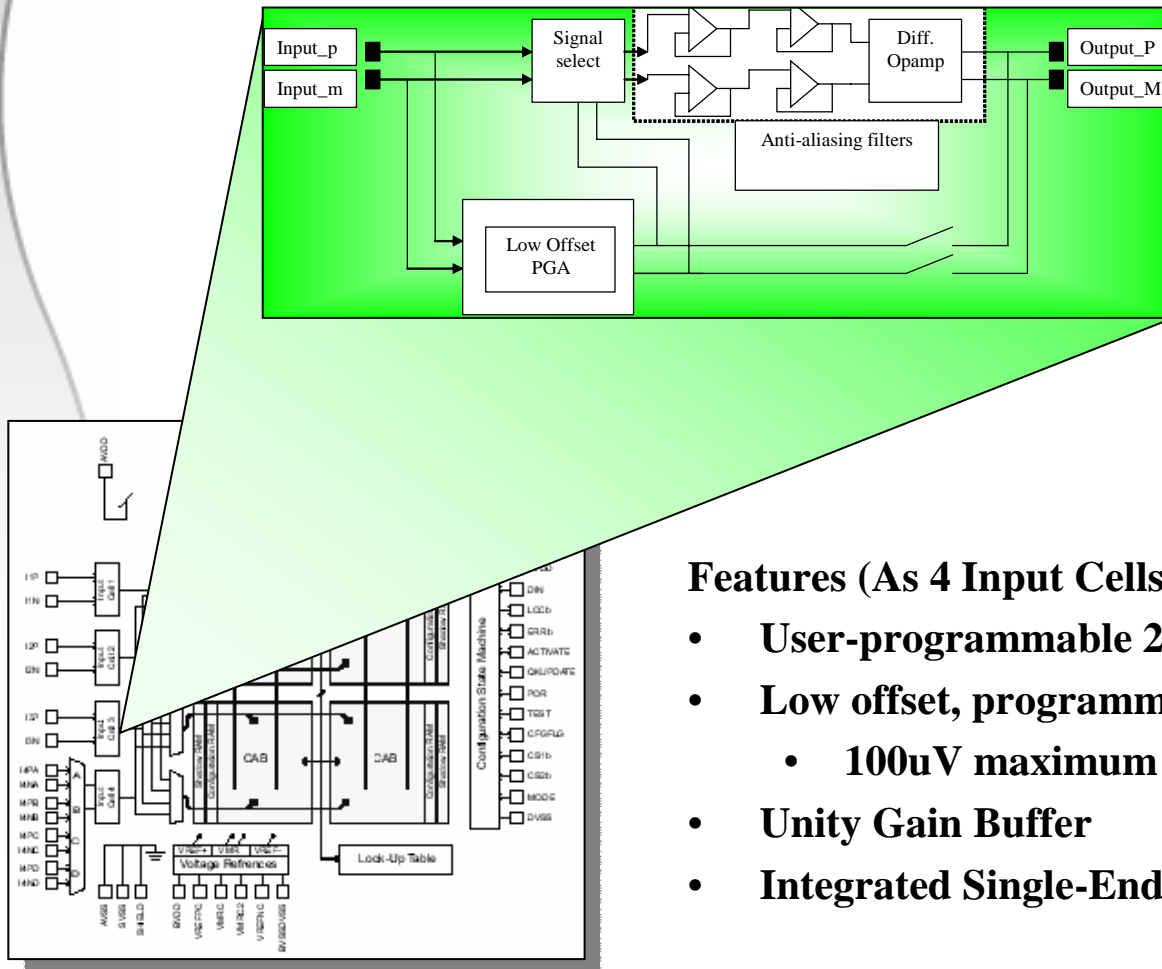
- **Programmable SC circuits offer:**
 - Flexible functionality
 - Real-time full or partial reconfiguration
 - Hardware-multiplexing

**COST SAVINGS AND PERFORMANCE
ENHANCEMENTS**



Anadigmvortex™ Silicon Overview

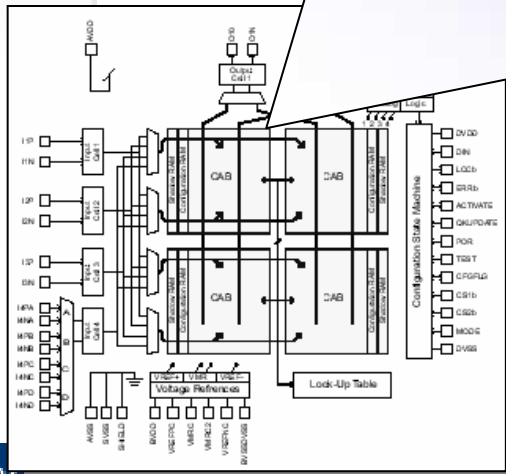
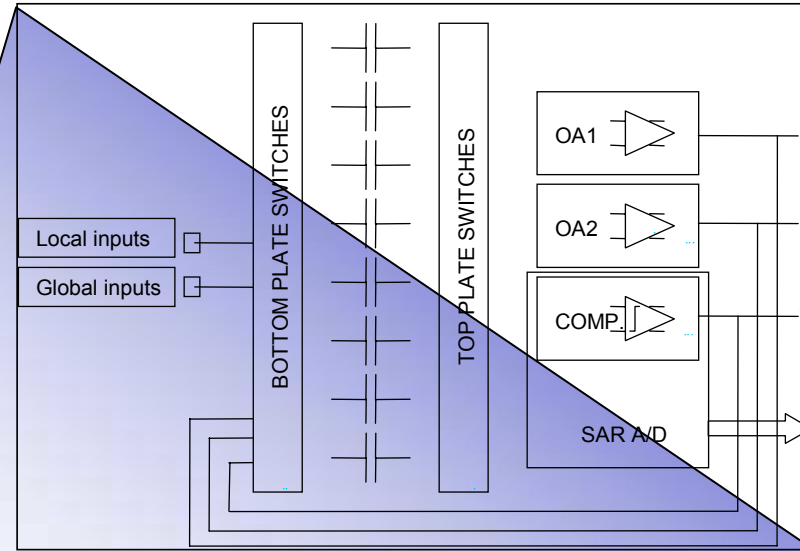
Input Cell



Features (As 4 Input Cells)

- **User-programmable 2nd order anti-aliasing filters**
- **Low offset, programmable gain amplifier (PGA)**
 - **100uV maximum offset (chopper mode)**
- **Unity Gain Buffer**
- **Integrated Single-Ended to Differential Converter**

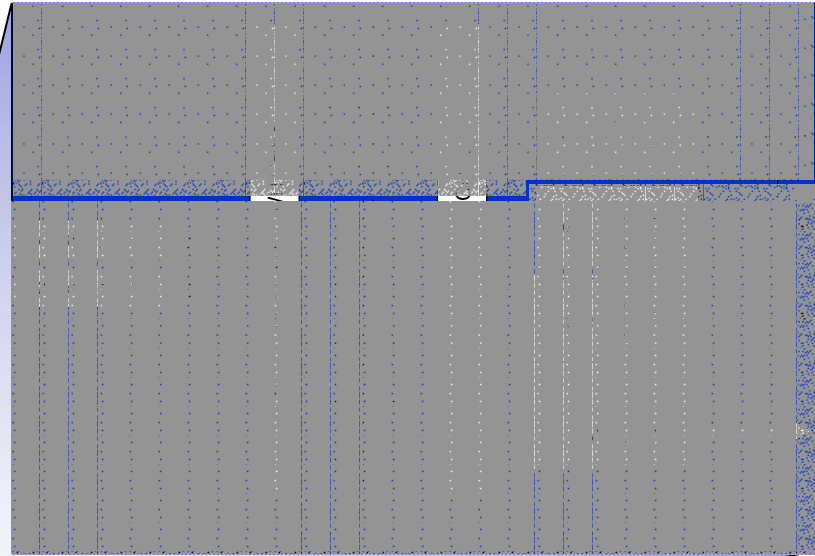
CAB (Configurable Analog Block) Cell



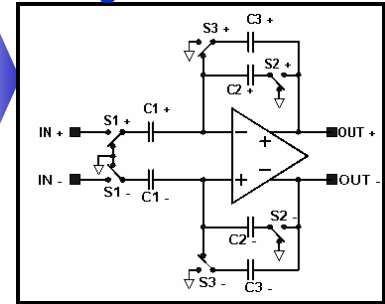
Features (x4 CAB Cells)

- 2 differential op-amps
- 1 differential comparator
- 1 SAR
- 8 switched capacitors
- Maximum Sample Rate=4MHz
- SNR = 80dB (108dB for BW=22kHz)

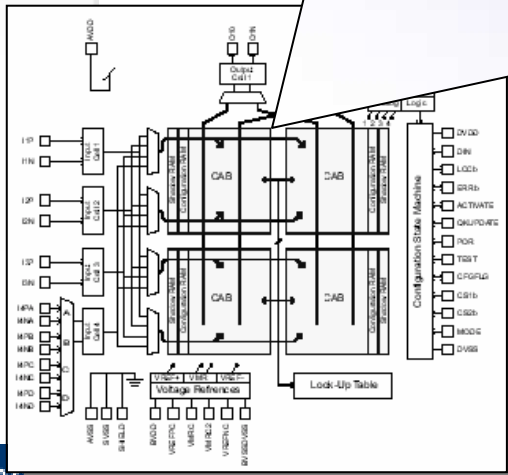
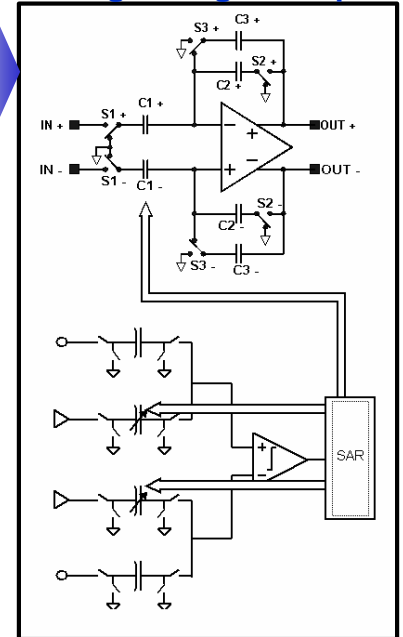
CAMs



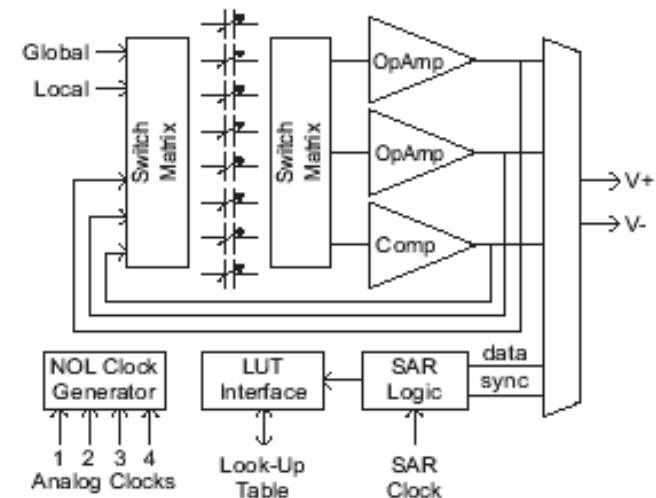
Programmable Gain



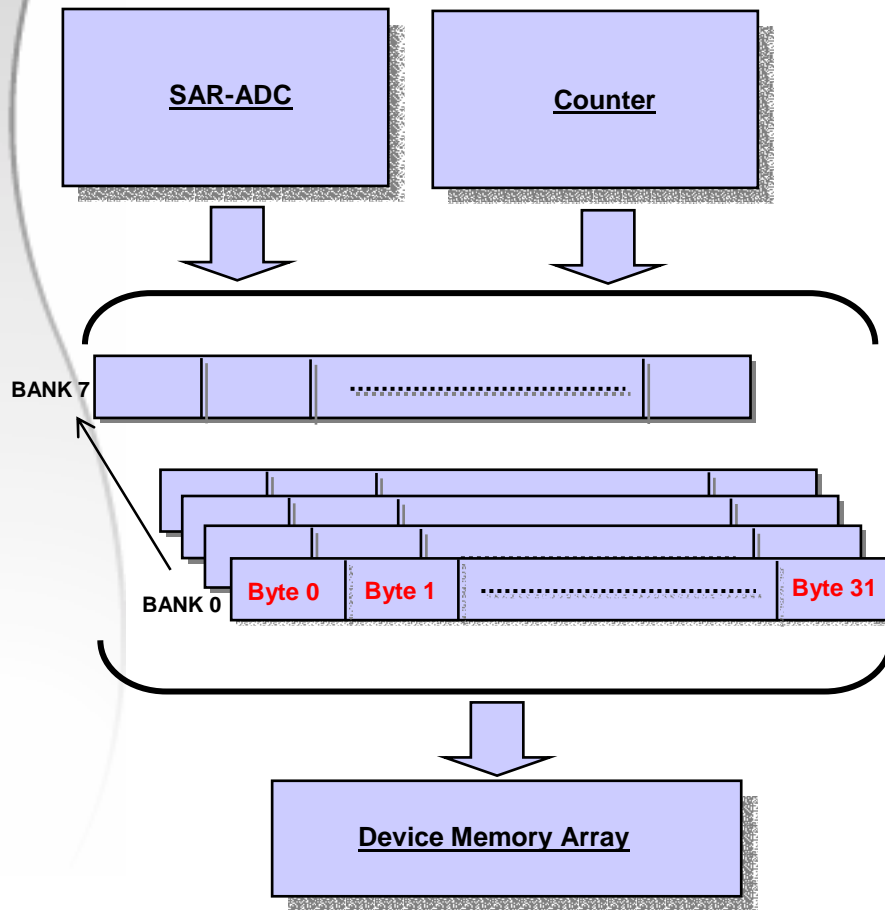
Analog Voltage Multiplier



- Each CAB has an 8-bit Successive Approximation Register (SAR) which can be used with the comparator to generate an Analog to Digital Converter (ADC).
- SAR-ADC requires 2 clocks
 - CLOCKA determines the successive conversion rate.
 - CLOCKB (=16 X CLOCKA) is used for the conversion itself.
- SAR Input is limited to VMR +/- 1.5V.
- SAR Output is routed to either:
 - LUT's address port
 - Host CAB



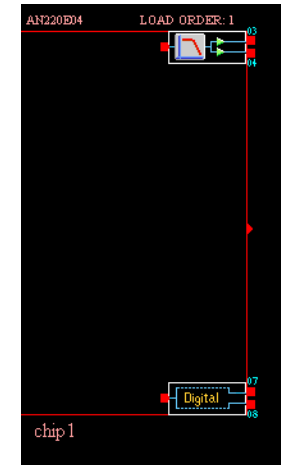
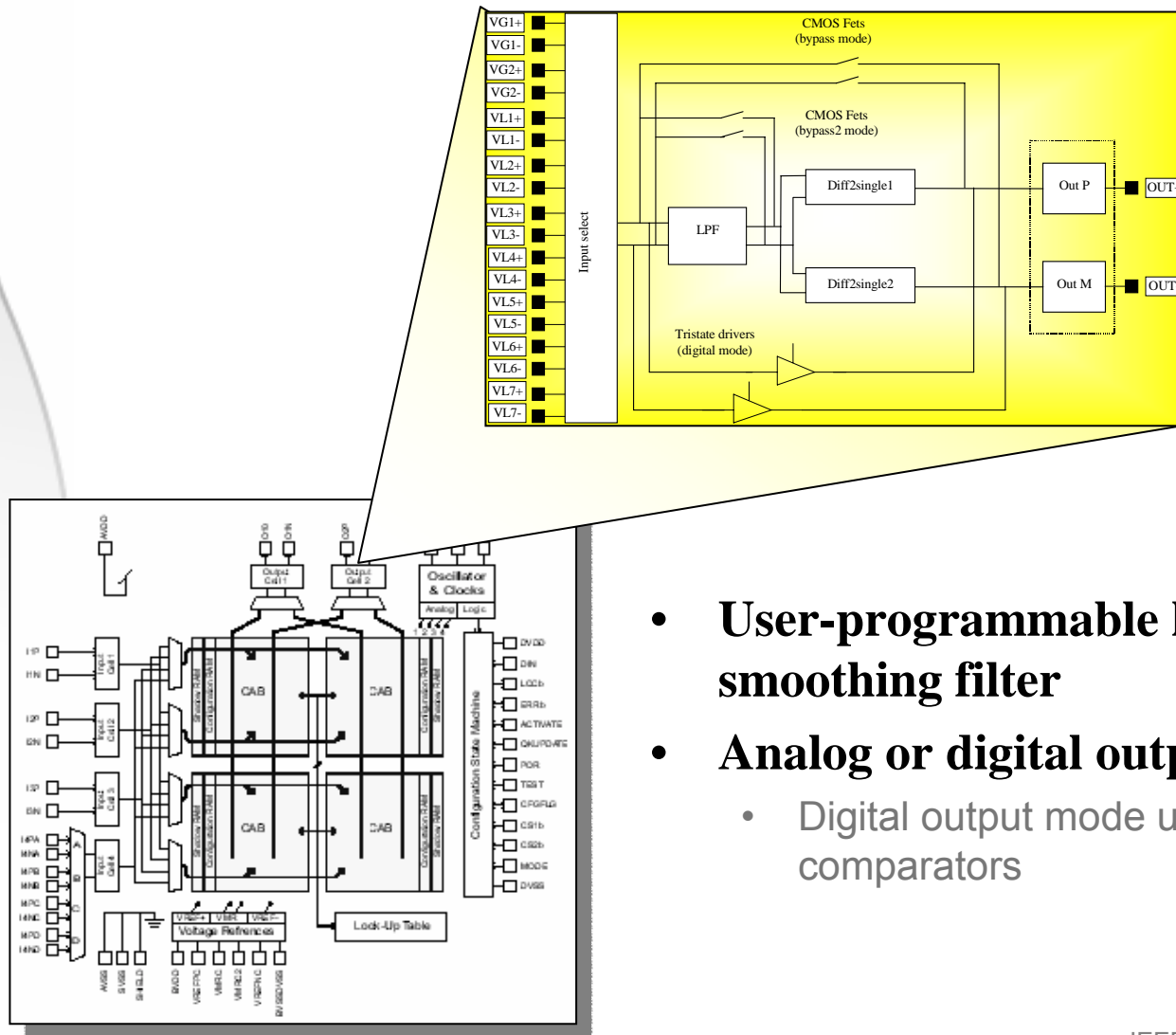
Look Up Table (LUT)



Features:

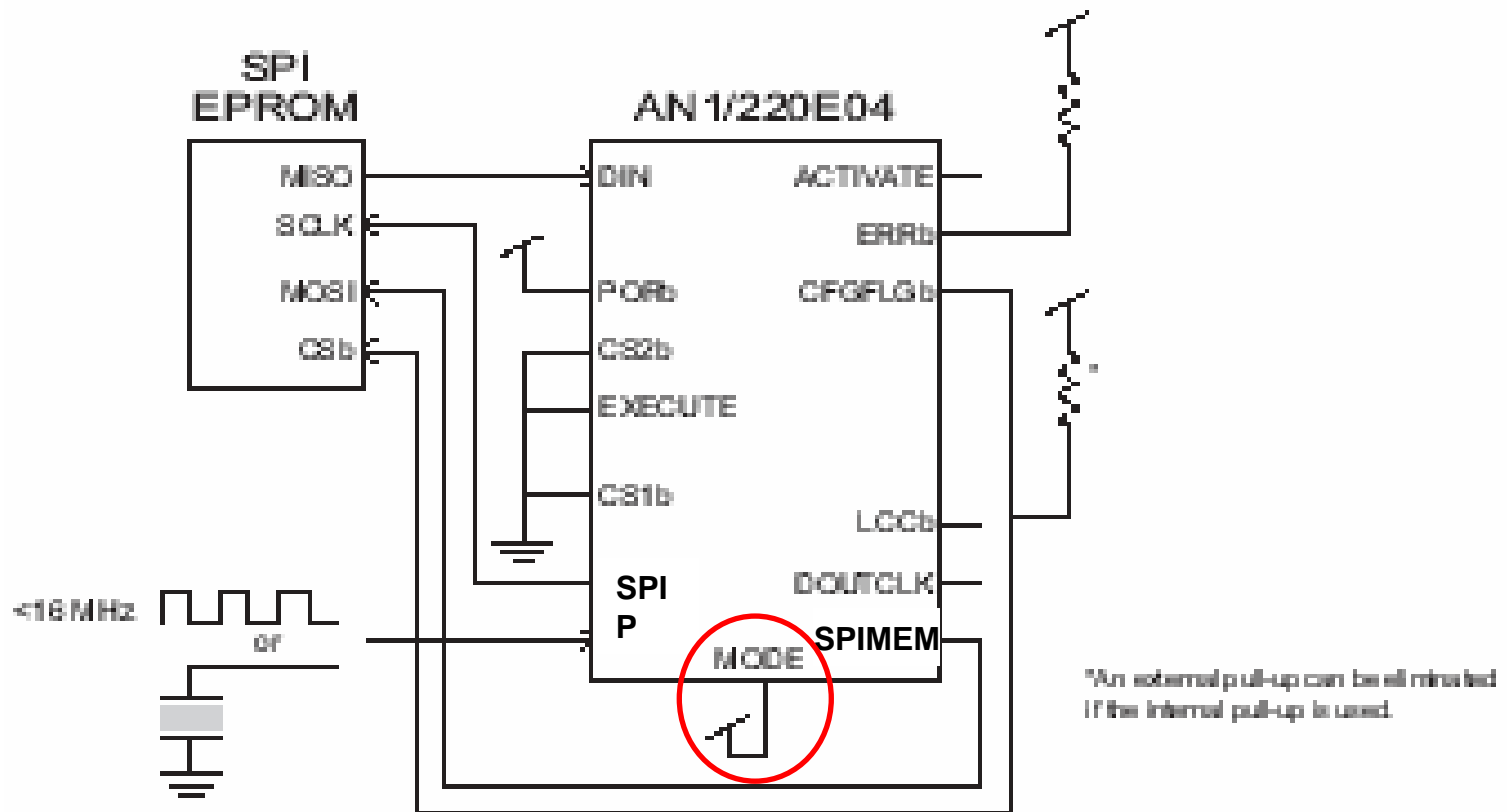
- **256-Byte SRAM LUT**
 - Eight banks of 32-bytes each
- **LUT can be addressed by any of the four SAR-ADC outputs OR a counter**
- **LUT's output can be used to change the operation of any part of the analog array.**

Output Cell



- **User-programmable low-pass smoothing filter**
- **Analog or digital output**
 - Digital output mode used with the internal comparators

Interfacing to a EEPROM*



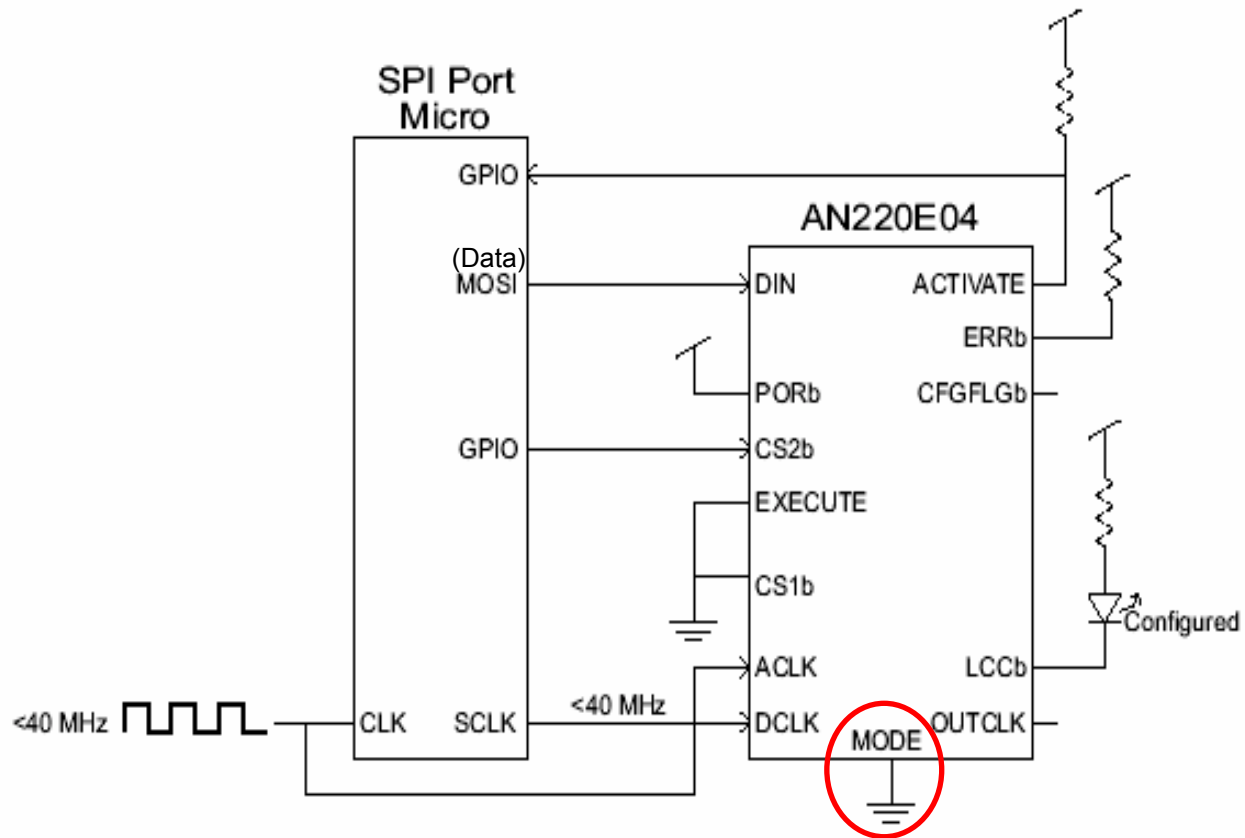
SPI – Serial Peripheral Interface

* 25 series SPI or 17 series EEPROM



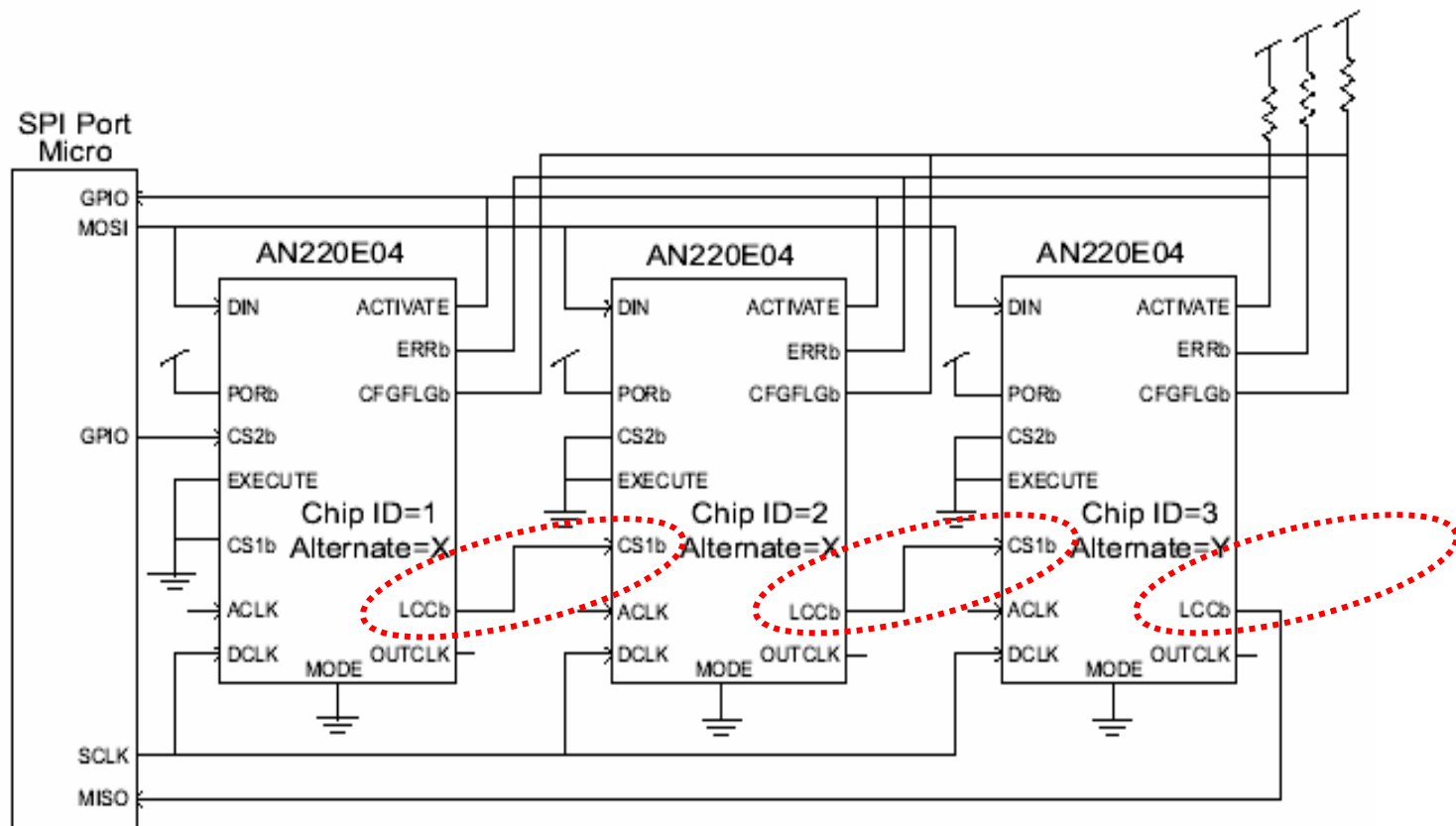
© Anadigm 2003. All rights reserved.

Interfacing to a Processor



SPI – Serial Peripheral Interface

Interfacing Multiple FPAsAs





AnadigmDesigner®2 Software Overview

AnadigmDesigner[®]2 – Overview

- **AnadigmDesigner[®]2**
 - Easy-to-Use
 - Intuitive “drag-and-drop” user interface
 - Built-in signal generator, oscilloscope
 - Built-in, accurate discrete-time behavioral simulator
 - Extensive help documentation

The screenshot displays the AnadigmDesigner2 interface. At the top, a window titled "Test generator and match filter 2 - AnadigmDesigner2" shows a circuit diagram with a signal generator and a filter. Below it, a waveform plot shows the signal's behavior. On the right, a control panel includes a "Display/Close" button, a "Main Pre-Delay" section with a "Delay" button and a "1.8 V" value, and a "Position" section with a "100 ns" value and a "Close" button. A "Help" window is open in the foreground, titled "Circuit Diagram and Design Equations:". It provides the transfer function for the circuit:

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{2\pi f_0 \frac{G}{Q} s}{s^2 + 2\pi f_0 s + 4\pi^2 f_0^2}$$

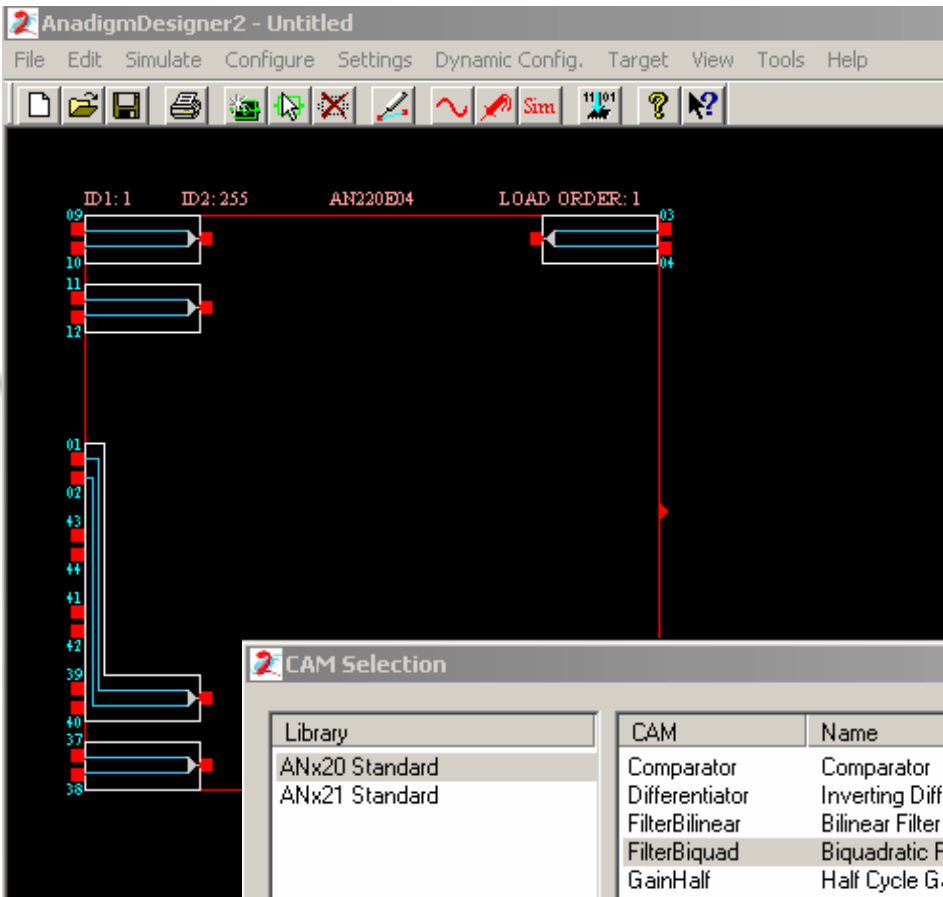
It explains that G is the pass-band gain (center frequency gain), f_0 is the corner (center) frequency, and Q is the quality factor. It also notes that the filter's gain is equal to $G |V_{in}|$ at the corner frequency, f_0 . The help window mentions that two different circuits can realize the band pass version of the biquadratic filter and that the implementation that best fits the combination of IPmodule Parameter values is selected. The Roman numeral in the upper right corner of the IPmodule symbol identifies the selected circuit (i or ii). The first circuit (1) realizing this band pass filter is shown in the figure below.

The schematic shows a complex circuit with two operational amplifiers. The first op-amp is configured as a non-inverting amplifier with a feedback network of capacitors C_1 and C_2 . The second op-amp is configured as an inverting amplifier with a feedback network of capacitors C_3 and C_4 . The circuit includes several switches labeled S_1 through S_8 and capacitors labeled C_1 through C_4 . The input is labeled $IN+$ and $IN-$, and the output is labeled $OUT+$ and $OUT-$.

The capacitor values are chosen based on the best ratios of the capacitors satisfying the following relations:

$$f_0 = \frac{f_c}{2\pi} \sqrt{\frac{C_3 C_4}{C_1 C_2}}$$

Main Screen



The CAM Selection dialog box is open, displaying a list of available CAM components. The dialog has a title bar with "CAM Selection" and standard window controls. On the right side, there are buttons for "Create CAM", "CAM Documentation", "Close", and "Help".

Library	CAM	Name	Approved
ANx20 Standard	Comparator	Comparator	Yes
ANx21 Standard	Differentiator	Inverting Differentiator	Yes
	FilterBilinear	Bilinear Filter	Yes
	FilterBiquad	Biquadratic Filter	Yes
	GainHalf	Half Cycle Gain Stage	Yes
	GainHold	Half Cycle Inverting Gain Stage with Hold	Yes
	GainInv	Inverting Gain Stage	Yes
	Hold	Sample and Hold	Yes
	Integrator	Integrator	Yes
	Multiplier	Multiplier	Yes
	OscillatorSine	Sinewave Oscillator	Yes
	PeriodicWave	Arbitrary Periodic Waveform Generator	Yes
	RectifierFilter	Rectifier with Low Pass Filter	Yes

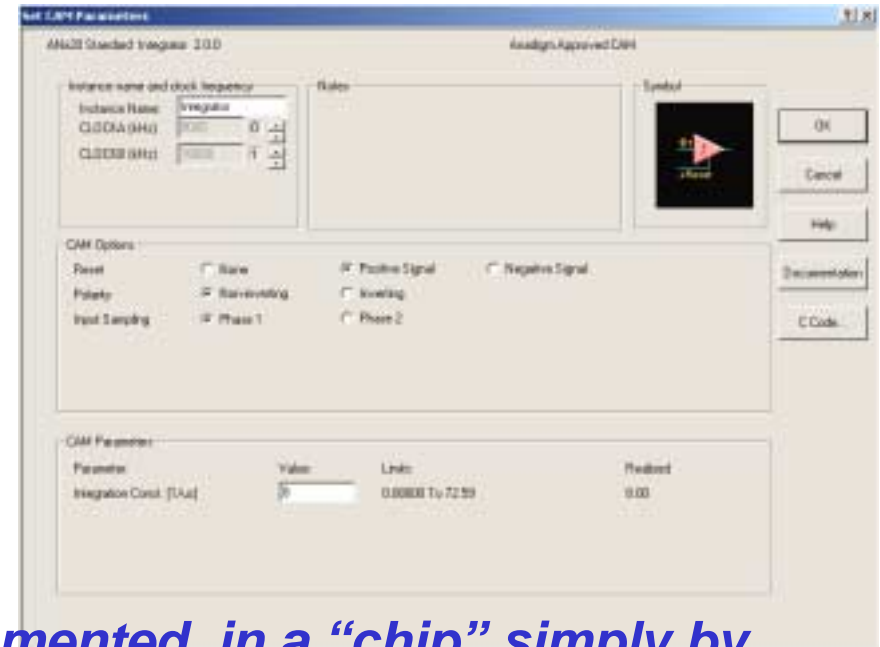
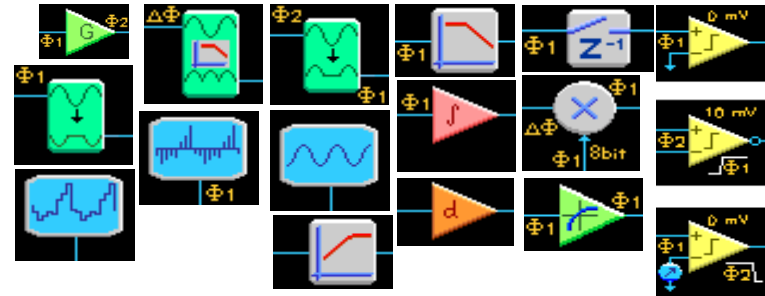
What are CAMs?

- **Configurable Analog Modules (CAMs)**

- Circuit building blocks abstracted to a functional level
 - Gain Stages
 - Summing Stages
 - Rectifiers
 - Filters

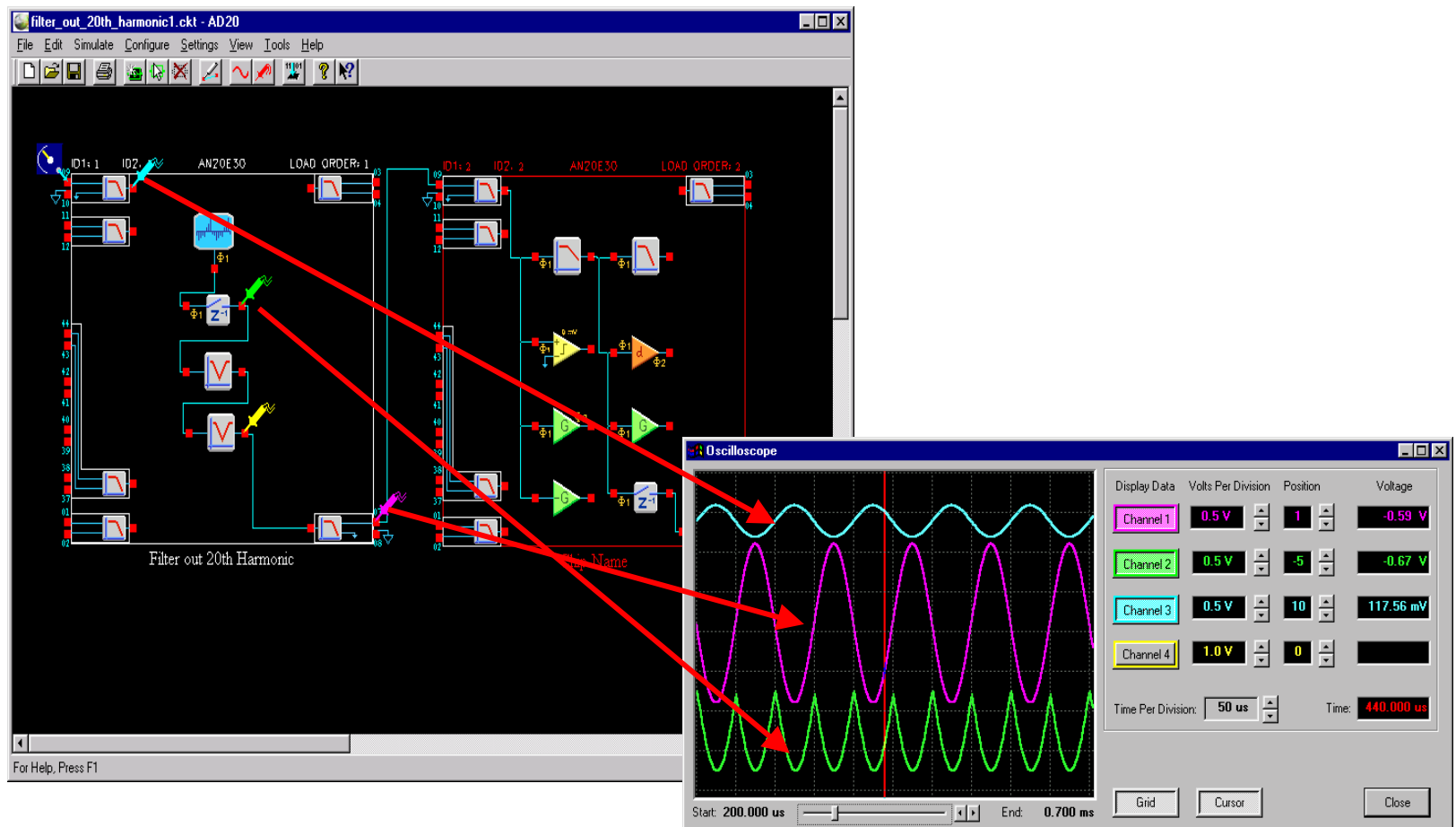
- **Each CAM is adjustable**

- Dynamic user interface – options and limits can change



A complex circuit can be implemented in a “chip” simply by selecting, configuring, placing and wiring CAMs

Simulating a Circuit



Using the AnadigmFilter™ tool

The screenshot displays the AnadigmFilter tool interface. The main window shows a circuit diagram with two filter stages, Filter0 and Filter1, each containing four stages (Stage0 to Stage3). The circuit is connected to an input and output. The right-hand panel shows the filter configuration parameters:

- Passband Ripple (0.01 to 9.99): 3.0 dB
- Passband Gain (-10 to +10): 0.0 dB
- Stop Band Atten. (5.0 to 80.): 30.0 dB
- Corner Freq. Low (0.01 Hz to 8 MHz): 1.5 kHz
- Stop Freq. Low (0.01 Hz to 8 MHz): 2.0 kHz
- Corner Freq. High (0.01 Hz to 8 MHz): 20.0 kHz
- Stop Freq. High (0.01 Hz to 8 MHz): 10.0 kHz
- Master Clock Freq. (100 kHz to 20000 kHz): 16780.0 kHz
- CAM Clock Freq. (1 Hz to 20000 kHz): 75.586 kHz
- Override CAM Clock Freq

The plot area shows a frequency response graph with a red curve representing the magnitude response. The x-axis is labeled 'kHz' on a logarithmic scale from 1 to 10. The y-axis represents gain in dB. A table below the plot shows the filter characteristics:

Poles	Chips
13	2
6	1
6	1
4	1

At the bottom of the interface, the current simulation parameters are displayed: Frequency = 0.0206 KHz, Amplitude = -0.000 dB, 1.000 W/V, Phase = -6.513 Deg, Group Delay = 0.317 mSec. The status bar also includes the text 'For Help, Press F1'.



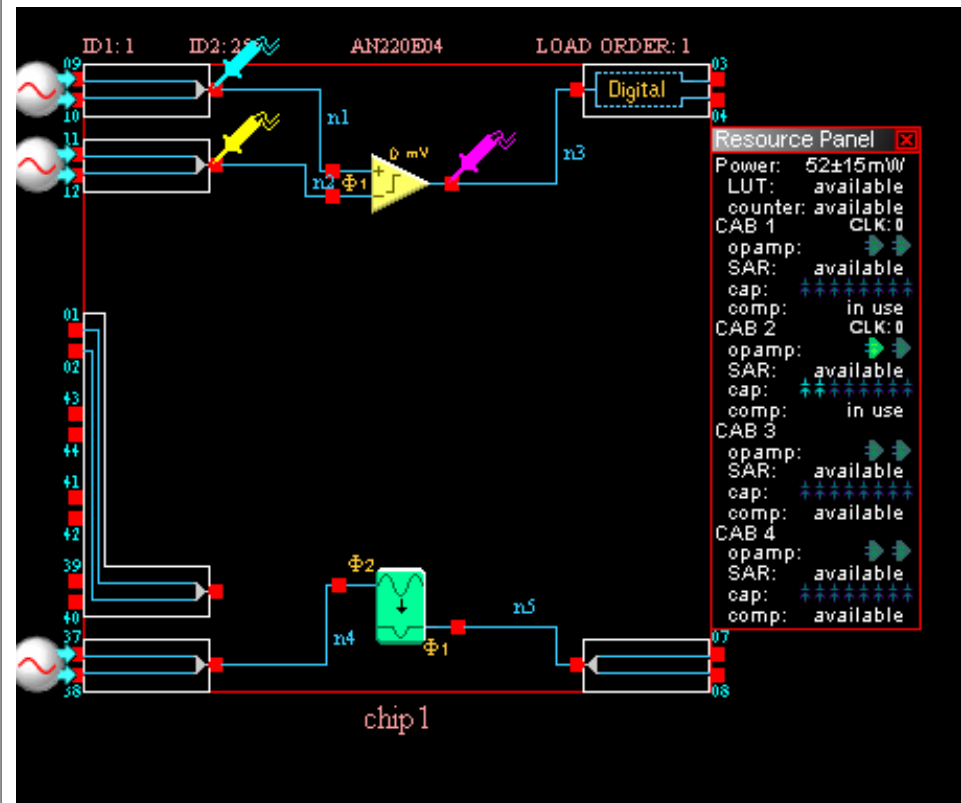
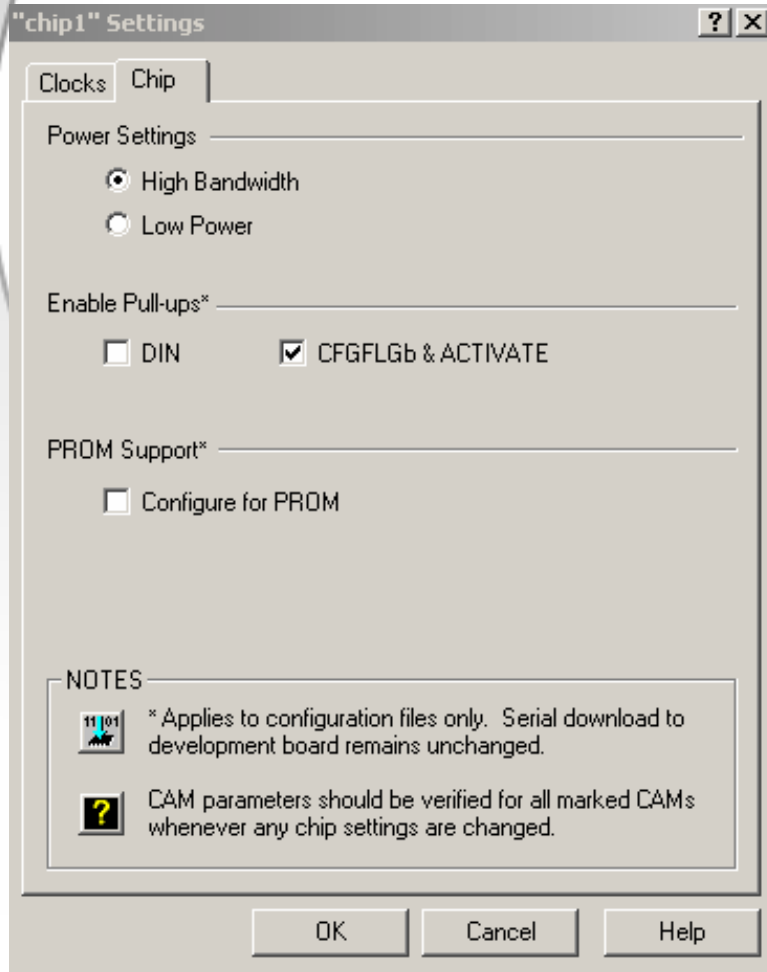
Clock Settings

The image shows the AnadigmDesigner2 software interface. The main window displays a circuit diagram for 'chip1' with various components and nodes. A 'Settings' menu is open, showing 'Active Chip settings...' and 'Preferences...'. The 'chip1' Settings dialog box is open, showing the 'Clocks' tab. The dialog box contains the following settings:

- Master Clock: Master Clock Frequency (fc): 16000.000000 kHz
- Clock Dividers:
 - Chopper Clock: $fc / 64 = 250.000000$ kHz
 - Sys Clock (fsys): $fc / 1 = 16000.000$ kHz
 - Clock 0: $fsys / 4 = 4000.000$ kHz
 - Clock 1: $fsys / 8 = 2000.000$ kHz
 - Clock 2: $fsys / 16 = 1000.000$ kHz
 - Clock 3: $fsys / 64 = 250.000$ kHz
- OUTCLK: Connect OUTCLK to: None
- NOTE: CAM parameters should be verified for all marked CAMs whenever any chip settings are changed.

Buttons: OK, Cancel, Help

Other Software Features





Static and Dynamic Reconfiguration

FPAA Configuration Downloads

- **From computer serial port to FPAA(s)**
 - On Anadigm Eval/Kit Board
 - All chips or Selected chips
- **From a file**
 - Formats
 - AHF, reversed AHF
 - Binary, reversed binary
 - Motorola S-1, reversed Motorola S-1
 - Motorola S-2, reversed Motorola S-2
 - For EPROM or Processor

Creating A Configuration File

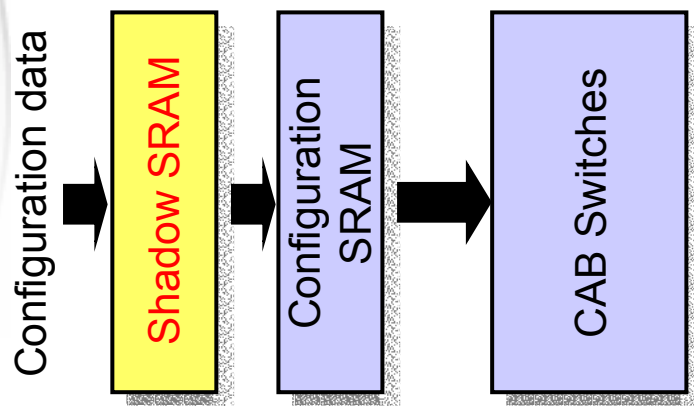
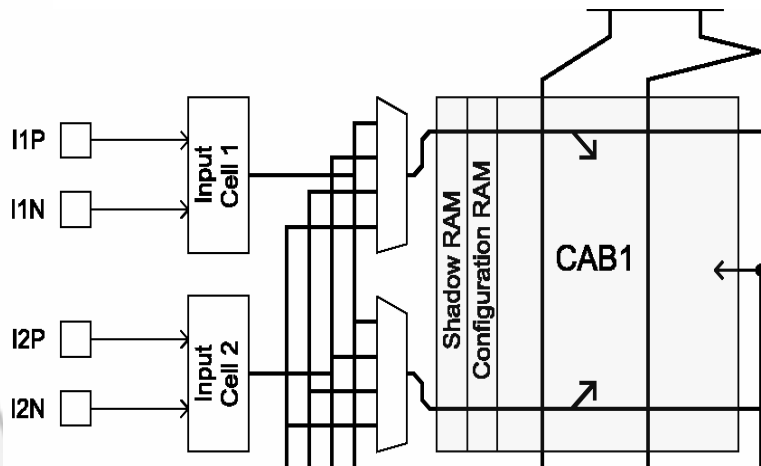
The image shows a software interface for creating a configuration file. The main window is titled "Designer2 - ToneGeneratorCircuit2" and has a menu bar with "Simulate", "Configure", "Settings", "Dynamic Config.", "Target", "View", "Tools", and "Help". The "Configure" menu is open, showing options like "Write configuration data to serial port" (Ctrl+W) and "Write configuration data to a ".abf" file".

Overlaid on this are three dialog boxes:

- Write Configuration Data:** A file save dialog with "Save in:" set to "Configuration Files". It shows two files: "Config File 1" and "Config File 2".
- Record Start Address:** A dialog with a "Start Address:" field containing "0" (Hexadecimal). It includes "OK" and "Cancel" buttons. A note below the field states: "(This value is normally zero for Serial PROMs). For 51 records, the maximum address is FC95".
- Config File 3 - WordPad:** A text editor window displaying a hex configuration file. The text is as follows:

```
S224000000000000000000000D5B72200300205CC000C2000A0040002007F0008FF012ADE00010FB9
S2240000202ACE013701483F483F012002FF2004FF0000483F483F300181300181190010CBD1
S22400004000100007C8002007000030018265002000660020000000081AB81AC2ACB030B90EB
S22400006000009800000000070419D2ADB031C9000900A80FF0010FF0181FF0010FF018100A7
S22400008007FF0182070000FF01282AC20503D8C1AE2ACB050B9800C1AC00000081CC009081
S2240000A02ADE05380A8001018126018197001097001000073C0182070000020182EE002093
S2240000C000EF002000480005AD81AC00000000080098000098080000419D81AC2ADB071777
S2240000E09000900A803F0010FC0010FC00100000000003000000072AC1090481AAC1AE2A24
S216000100CA0907419D000000000812A9E09020A802A0028
S9030000FC
```

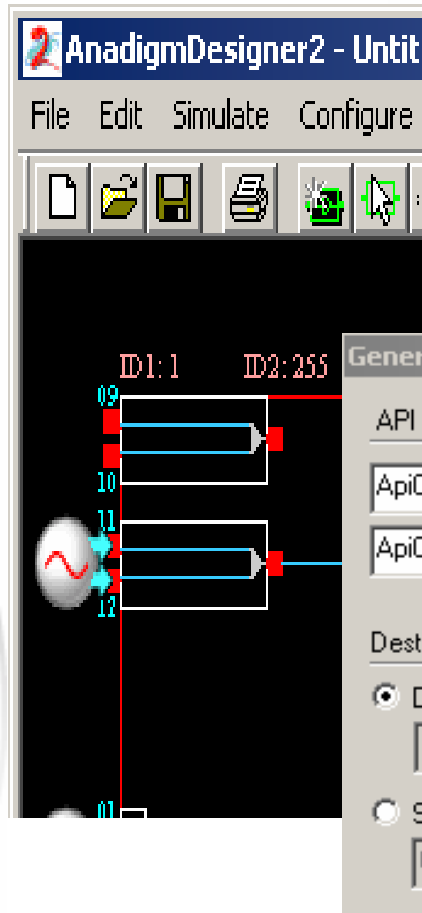
Dynamic Reconfiguration



Features:

- **Each CAB has two RAM blocks**
 - Configuration SRAM
 - Shadow SRAM
- **After power-up, data moves from the outside world**
 - To the shadow SRAM
 - Then to the configuration SRAM
- **While the device is operating**
 - The contents of the shadow SRAM can be updated without impacting device operation
- **Shadow SRAM contents can be transferred to configuration SRAM 'on-the-fly'**

Dynamic Reconfiguration - C-Code



```

CAMCode - WordPad
File Edit View Insert Format Help

| Description
| -----
| This function sets the 3dB corner frequency of the filter in
| this Output cell.
| -----
| Instance Name      nCAM          nChip
| -----
| OutputCell2       an_chip1_OutputCell2 an_chip1
| -----
|*-----*/

/*#####*\
#
#          Comparator.cam          #
#
\#####*/

/*-----*\
)          Hysteresis and Output Polarity  (
\-----*/

/*-----*\
|          CompSetOutputPolarity
+-----+
|
| Function Declaration
| -----
|* void an_CompSetOutputPolarity(an_CAM nCAM, an_Comparator NewPol);
/* -----*\
|
| Description
| -----
| This function controls the output polarity of the Comparator.
| -----
| Instance Name      nCAM          nChip
| -----
| Comparator1       an_chip1_Comparator1 an_chip1
| -----
|*-----*/

/*#####*\
#
#          RectifierHalf.cam          #
#
\#####*/

/*-----*\
)          Gain
\-----*/

/*-----*\

```

? X

enerate

ose

elp

n Options...

nctions...



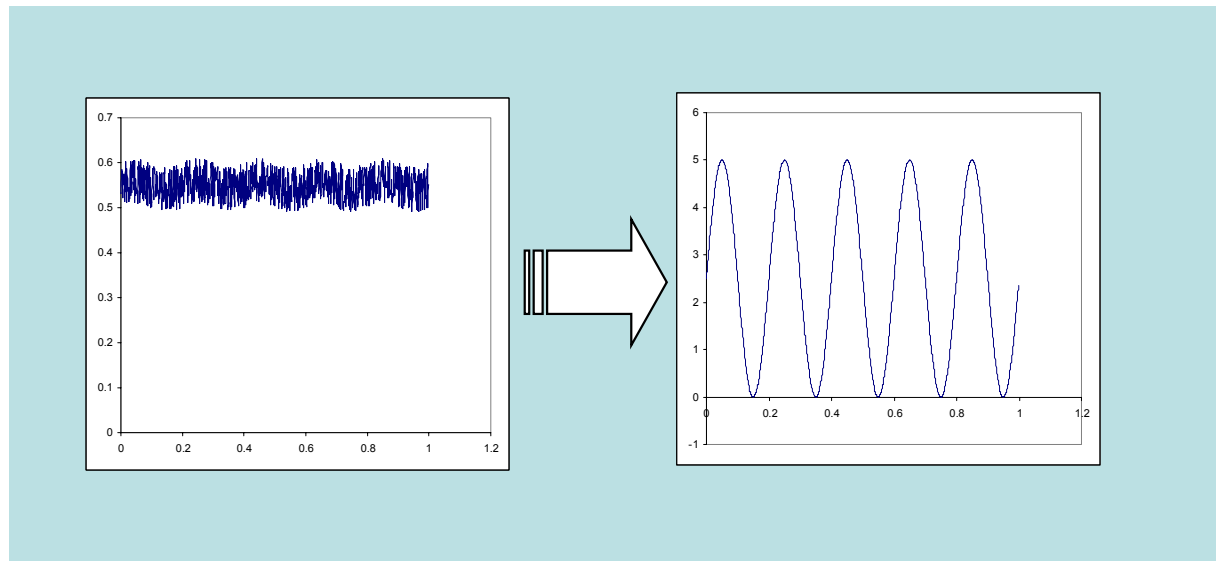


Measurement & Sensing Applications

Measurement & sensing applications

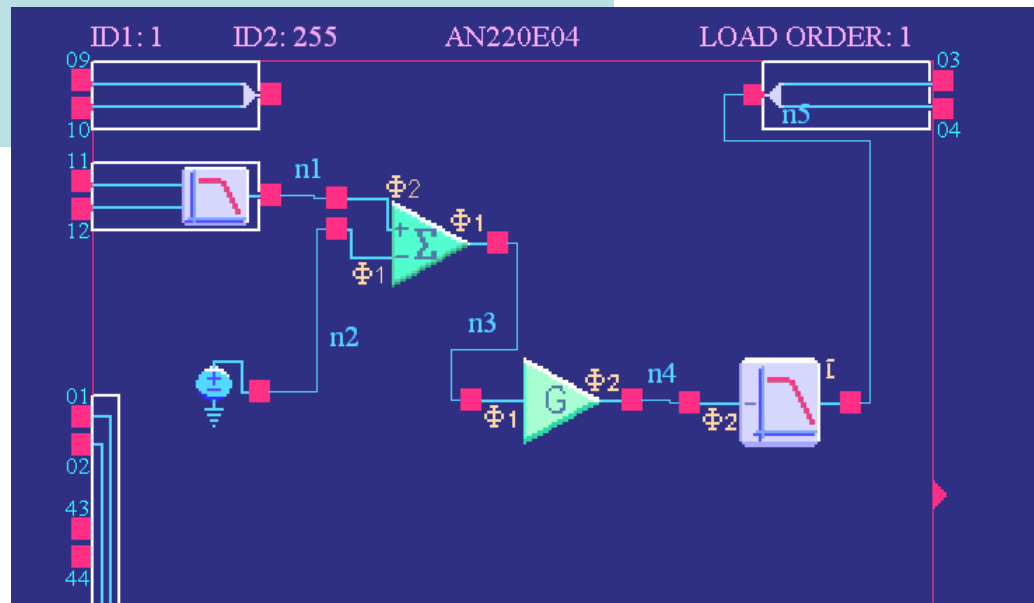
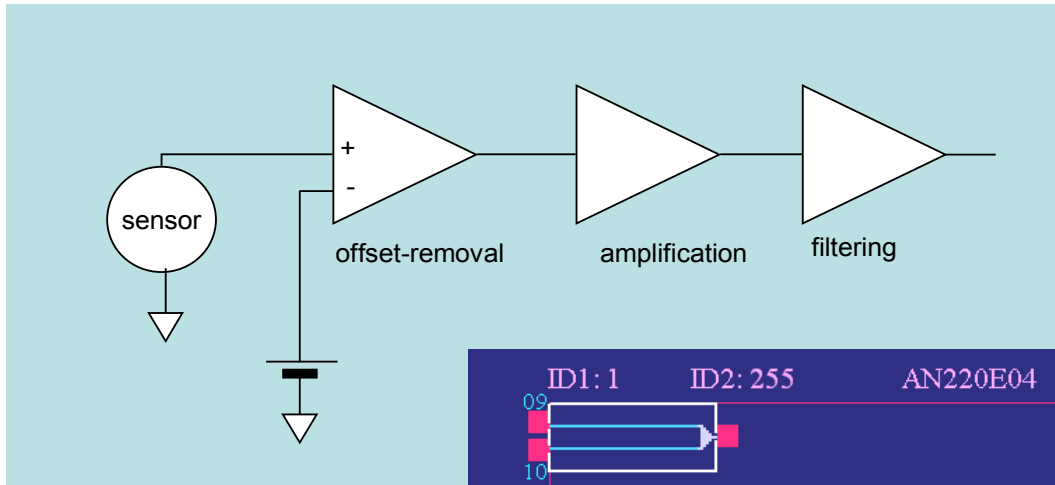
- **Common signal conditioning tasks:**

- Amplification
- Offset removal
- Rectification
- Filtering



Measurement & sensing applications

- **Common conditioning tasks**



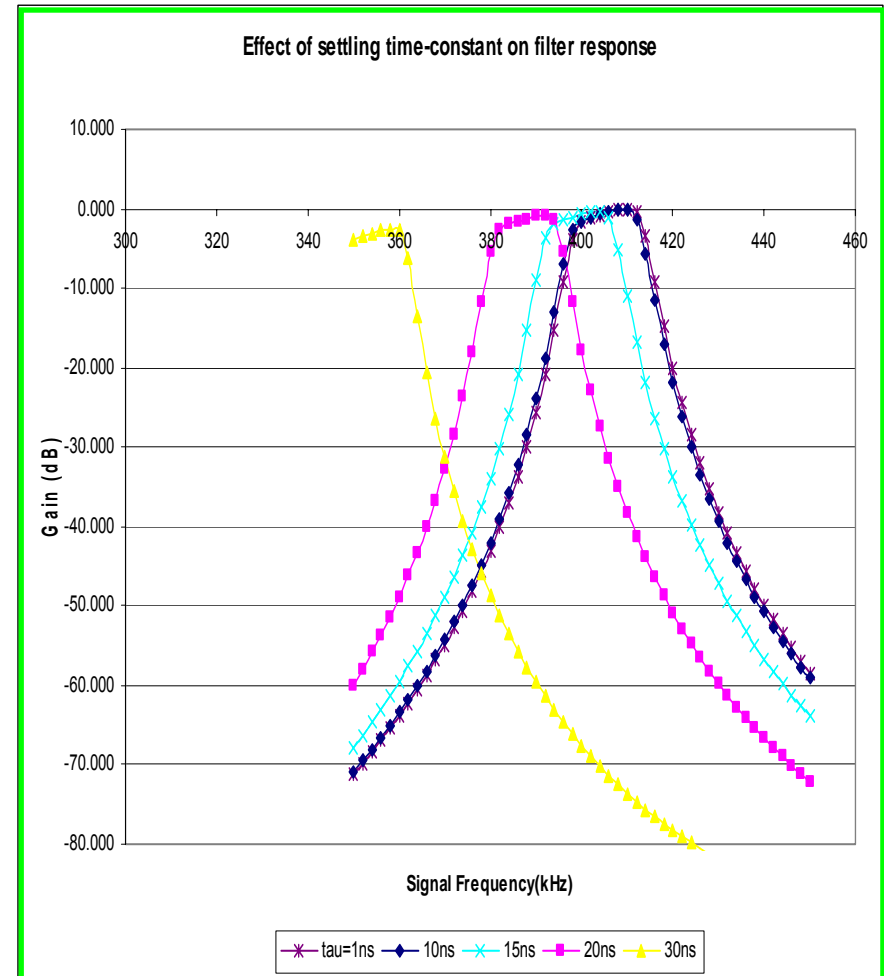
Operating limits (1): high clock freqs

- **Finite OpAmp Bandwidth**

- Capacitors must charge or discharge within each half clock-cycle
- OpAmp speed affects settling-time

- **Non-zero switch resistance**

- Similar effect on settling-time



Operating limits (2): high signal freqs

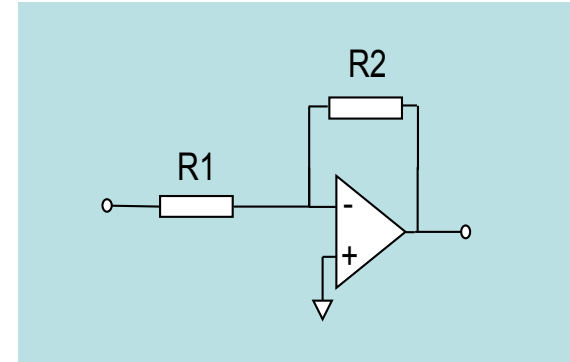
- **Finite OpAmp Gain**

- Provided opamp has high *open-loop* gain and high input-resistance then:

$$\frac{1}{G} = -\frac{R1}{R2}$$

- If open-loop gain is not infinite then:

$$\frac{1}{G} = -\frac{R1}{R2} \left(1 + \frac{1}{A} \right) - \frac{1}{A}$$



Operating limits (3): low clock freqs

- **“Droop”**

- The charge that should be stored on a node is:

$$Q = C.V$$

- This charge will be changed

(+ or -) by leakage:

$$\Delta Q_L = I_L.T$$

- So, the error will be

$$\text{Error} = I_L.T / C.V$$

- For a specified max error:

$$T < \frac{(C.V)_{\min} \cdot \text{Error}}{(I_L)_{\max}}$$

So, using some approximate values:

- $C \cong 10^{-11}$ (farad)
- $V \cong 10^{-3}$ to 1 (volt)
- So, $CV \cong 10^{-11}$ to 10^{-14} (coulomb)
- $I_L \cong 10^{-15}$ to 10^{-18} (amperes)

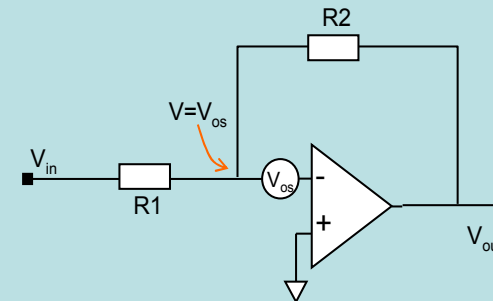
For a leakage errors of 0.1% or less:

$$T < (C.V)_{\min} \cdot \text{Error} / (I_L)_{\max} \cong 10^{-14} \cdot 10^{-3} / 10^{-15} \cong 10^{-2} \text{ (secs)}$$

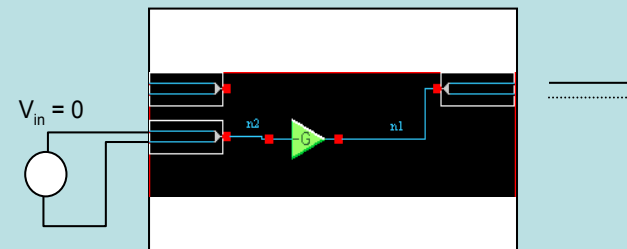
We can obtain leakage errors of 0.1% or less for millivolt signals, for sampling frequencies in the low kHz range.

- **Offset errors**

- Slight mismatches inside the opamp require balancing with a small input voltage
- Output voltage will have a DC error

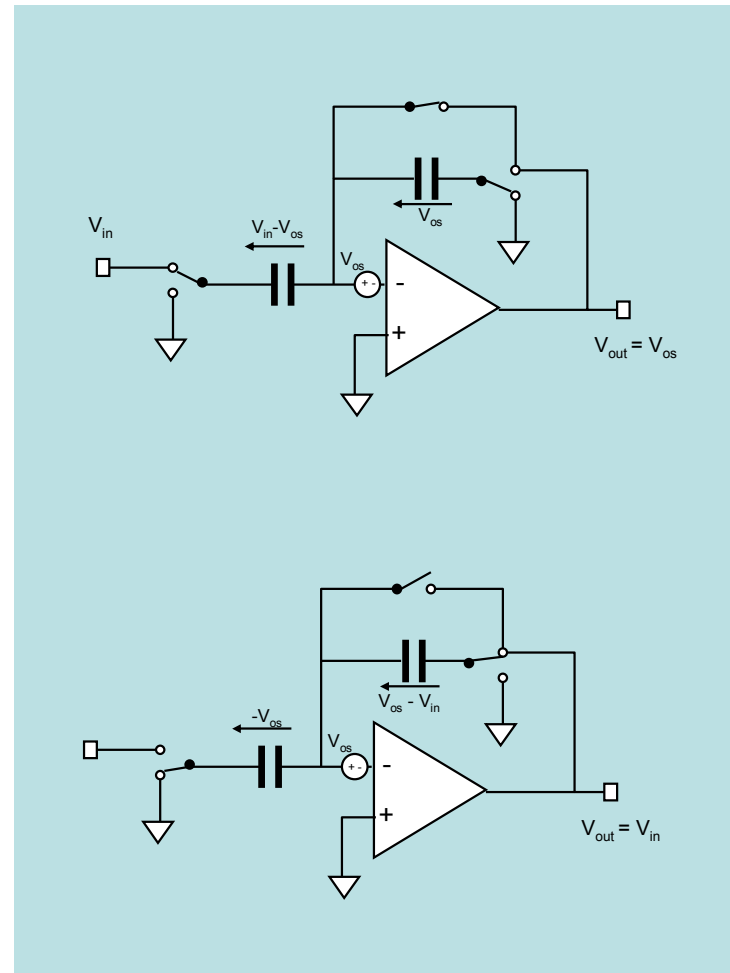


$$V_{out} = -V_{in} \cdot \frac{R2}{R1} + V_{os} \left(\frac{R2}{R1} + 1 \right)$$



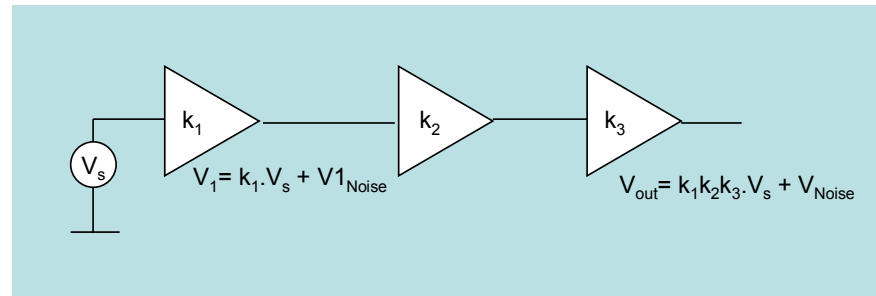
Offset compensation

- **“Half-cycle” building blocks allow cancellation of offset voltage**
 - GainHalf
 - GainHold
 - RectifierHalf
 - RectifierHold
 - SumDiff
- **“Offset-compensated” in one clock phase only (return-to-zero or hold-with-offset in other phase)**
- **Gain-Bandwidth Product (GBP) and slew-rate are not infinite!**
 - Selection of high-gain will limit max clock frequency (and vice versa) of RTZ CAMs



Achieving high-gain

- **Cascade stages – high-gain at front-end for low-noise**

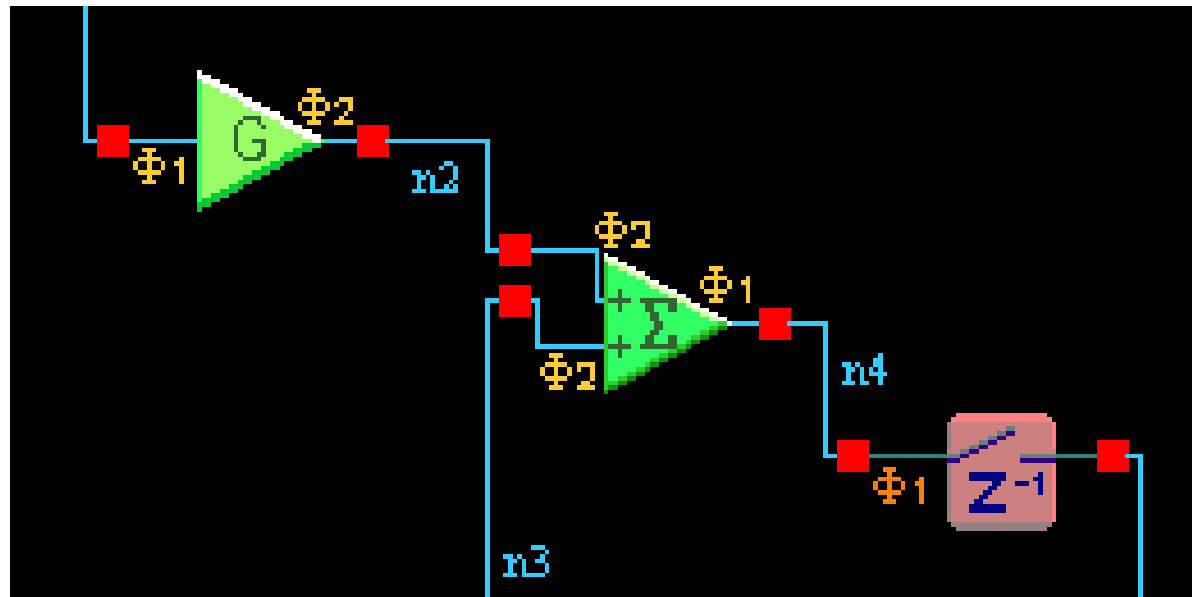


$$V_{noise} = K \cdot \left(\frac{V1_{noise}}{k_1} + \frac{V2_{noise}}{k_1 k_2} + \frac{V3_{noise}}{k_1 k_2 k_3} \right)$$

- **Use offset-compensated CAMs when implementing large gains**
 - Remember GBP limitations – select appropriate clock frequency (but watch anti-aliasing requirements)

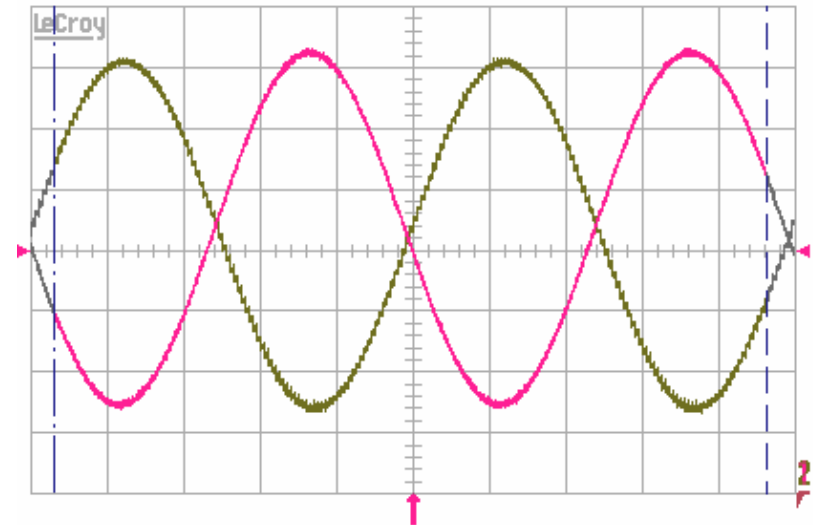
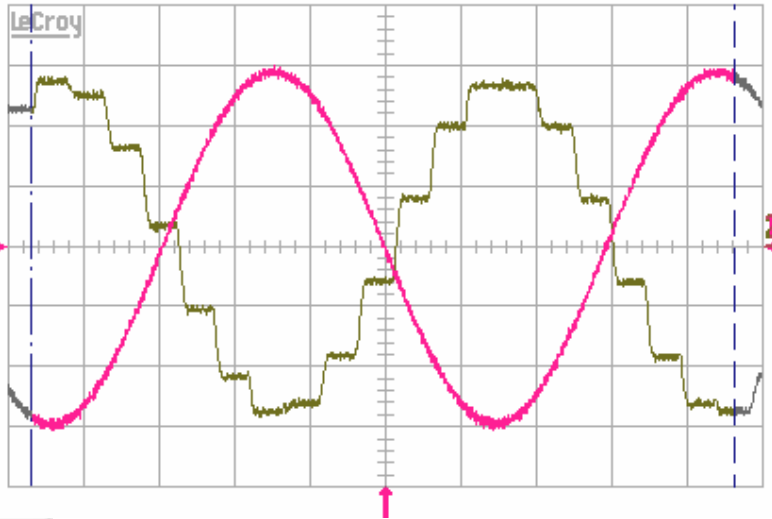
Phase selection

- Some CAMs have outputs which are only valid (or offset compensated) in one phase
- Make sure input sampling and output valid phases match!



Output Waveforms (1)

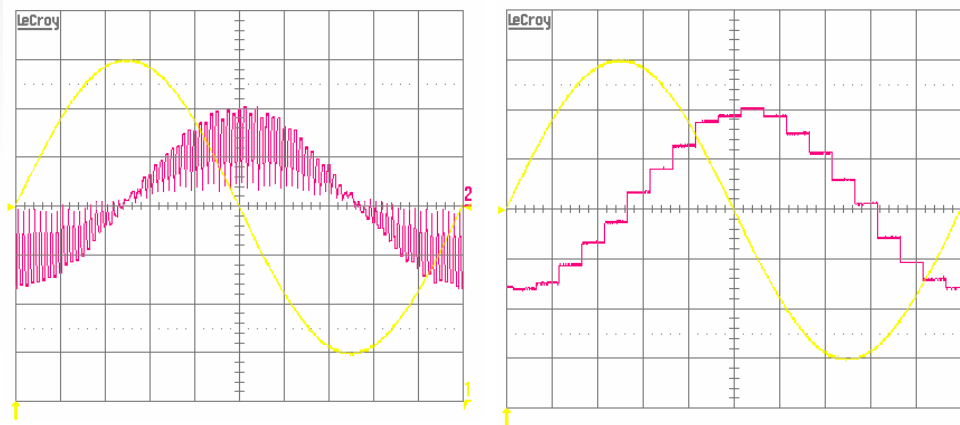
- **Staircase outputs:**



- Use high $f_{\text{clock}}/f_{\text{signal}}$ ratio to reduce step size and clock noise
- Keep $f_{\text{corner}}/f_{\text{signal}} > 30$ if using output cell filter
- If output is being sampled (eg by ADC), steps may be desirable !

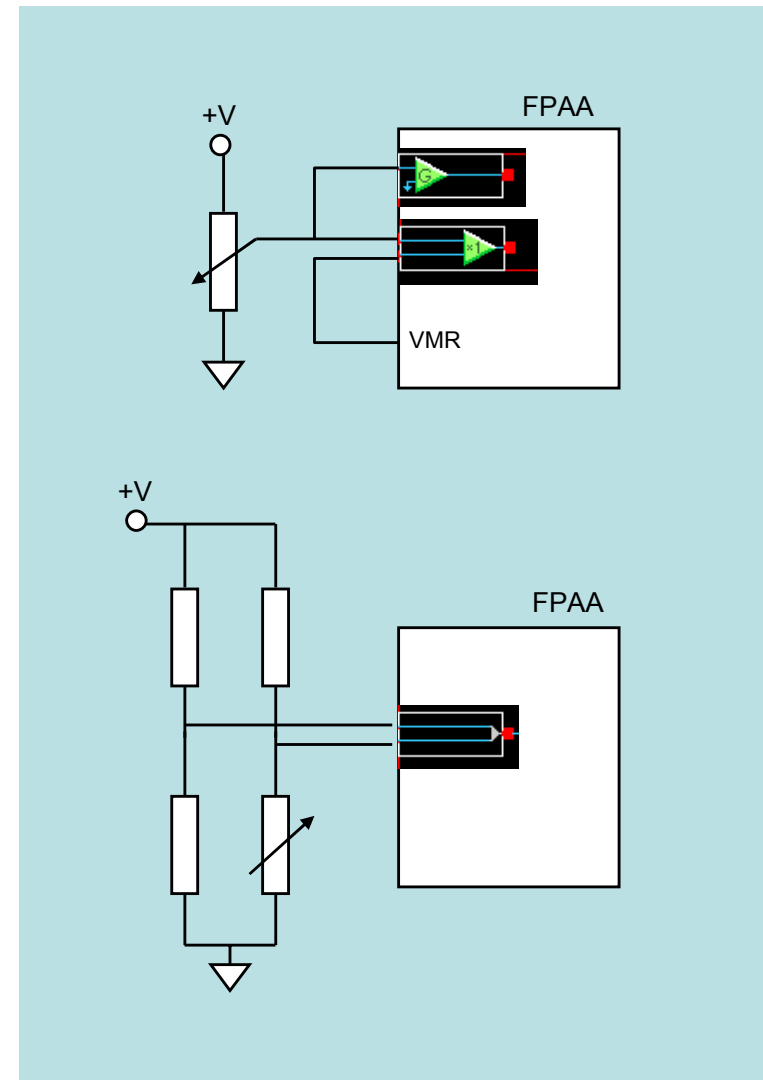
Output Waveforms (2)

- **“Final-value” outputs**
 - Differentiator (and transimpedance amplifier) outputs only reach their final value at the end of a clock phase
 - Final-value CAM outputs need capturing by CAMs which sample in one-phase and produce a valid output in the next



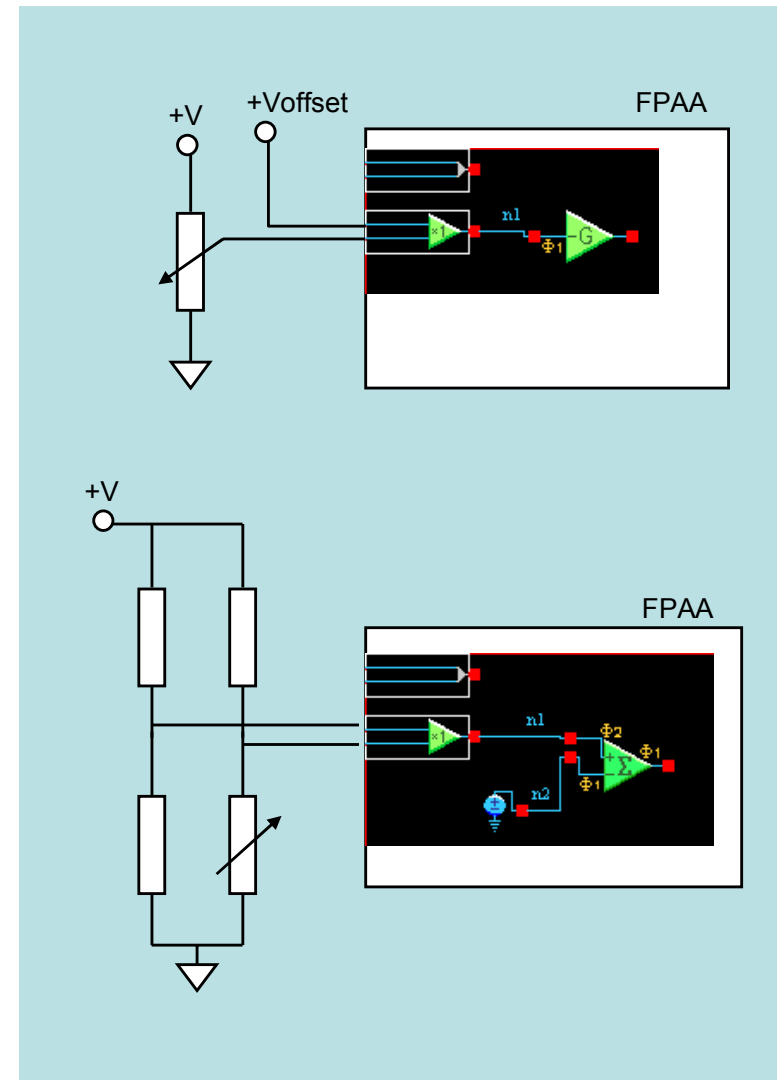
- **Hold**
- **Non-inverting GainHalf**
- **Non-inverting SumDiff**
- **Non-inverting SumFilter**

- **Internal signal ground is 2 volts (VMR)**
 - Centre inputs on 2V if possible (use Wheatstone bridge?)
 - Input range 0 – 4V
 - Convert external single ended signals to internal differential signals for max dynamic range



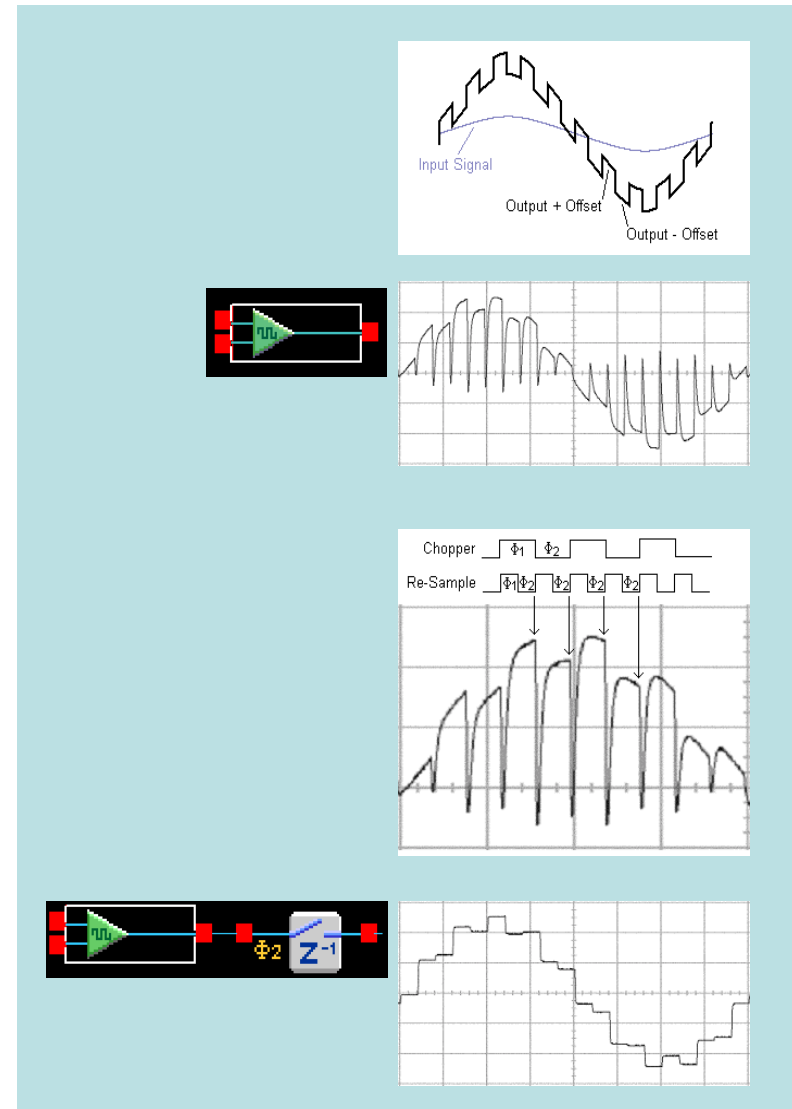
Offset removal

- **Single ended signals**
– apply offset to 2nd input pin
- **Differential signals –**
apply offset internally
- **Using differential signals gives better CM noise immunity**



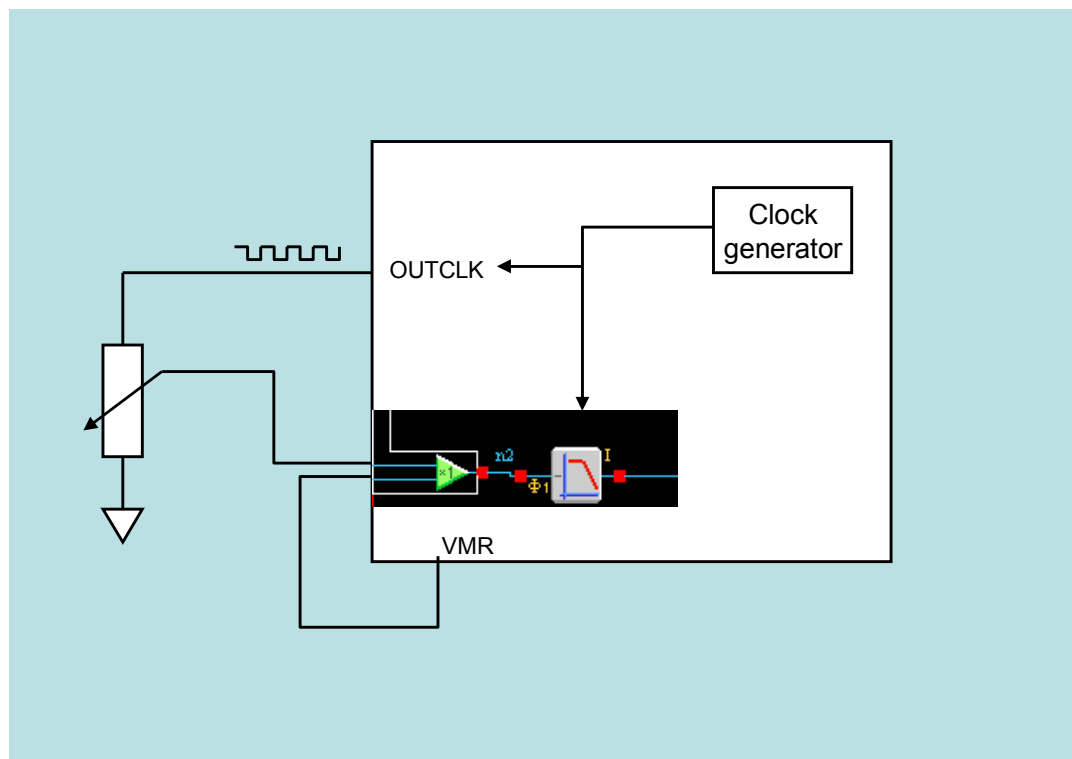
Using the chopper input

- **“Chopping” modulates the input signal with a square wave**
 - Low freq noise and DC offsets are pushed out to the clock freq (for easy filtering)
- **Slewing and settling limitations mean the chopper output is not “ideal”**
 - To recover the “ideal” output, sample in phase2 *at twice the chopper frequency*
- **To boost the gain, use offset-compensated CAMs, *sampling in phase2 at twice the chopper frequency***



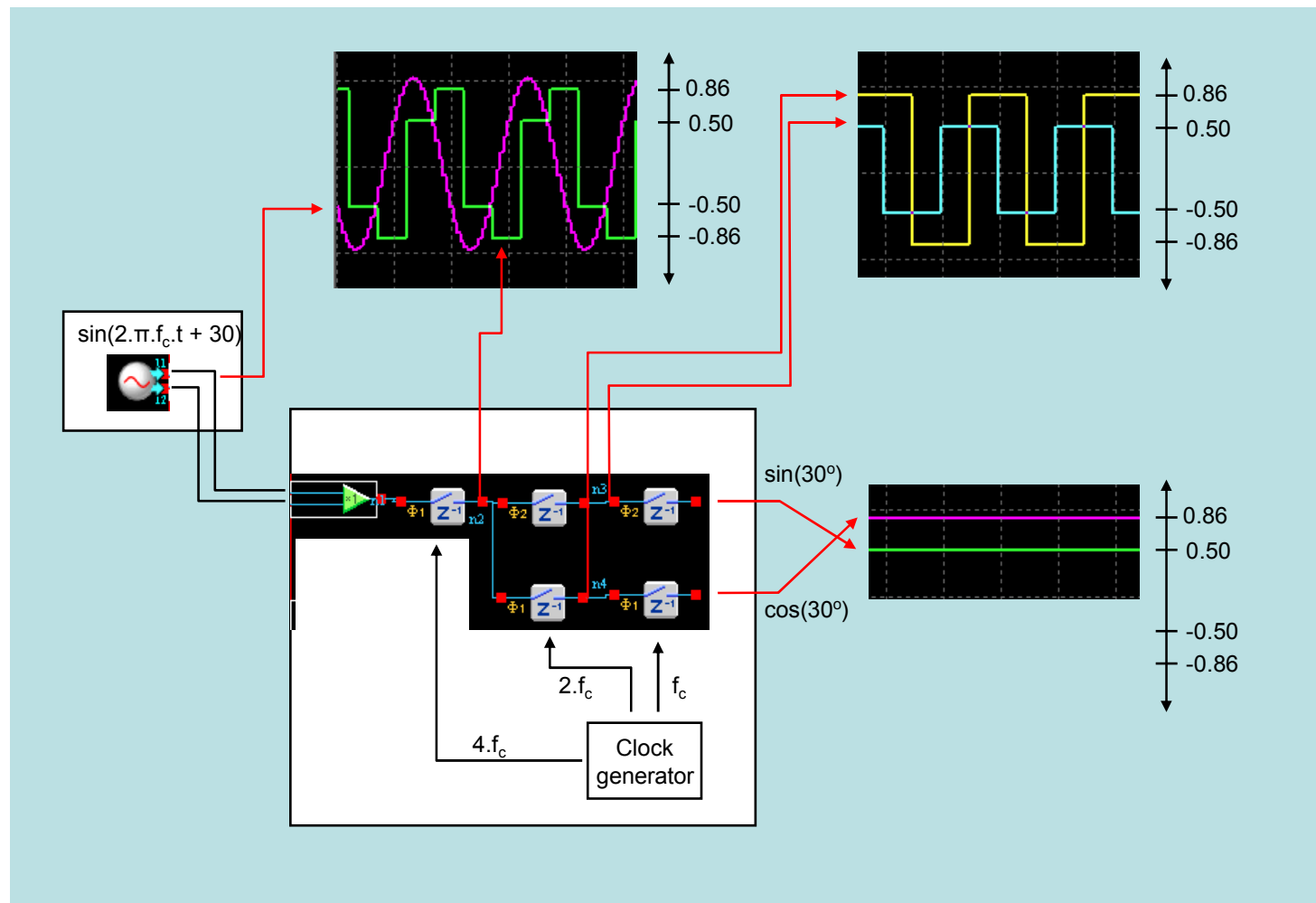
Noisy environments

- Use differential signals where noise can be made common-mode
- Use modulated sensor stimulus and synchronous demodulation (lock-in detection)



Phase detection

- Synchronous demodulation



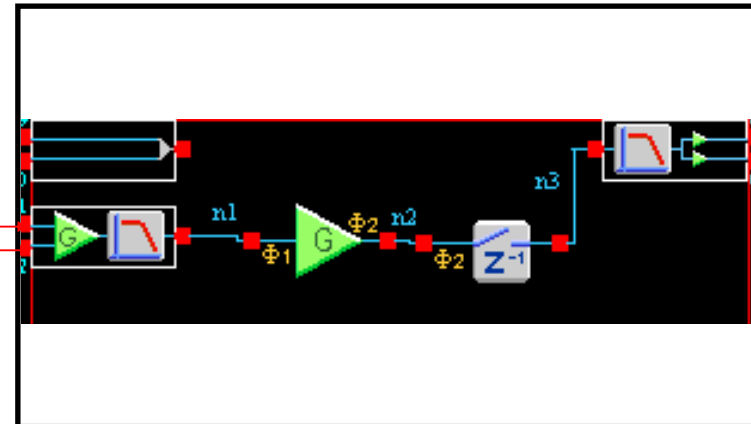
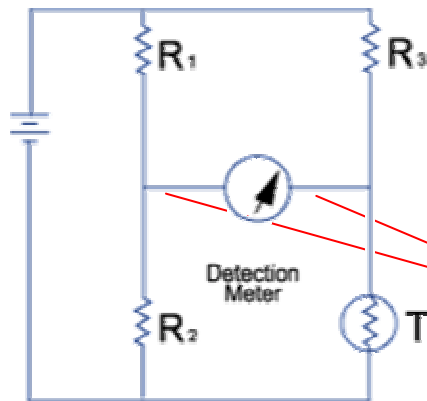
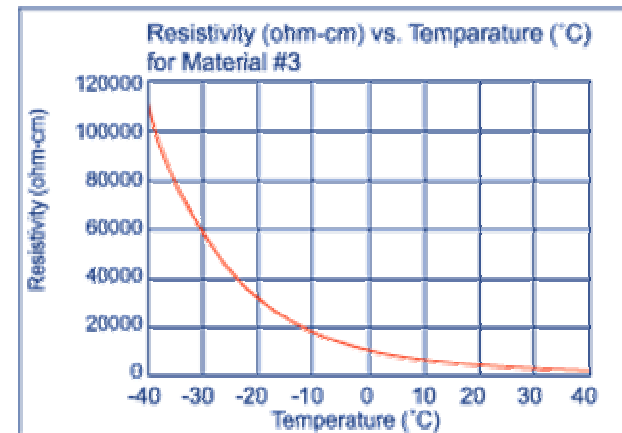
Anti-aliasing & clock selection

- **Input signal should be band-limited to $\ll f_{\text{clock}}/2$**
 - prevents hf noise adding to lf noise
 - Use a clock *at least* 5–10 times faster than the maximum signal
- **Use continuous-time not SC filtering**
 - attenuation at $f_{\text{clock}}/2$ should attenuate any hf signals by the required SNR
 - Keep $f_{\text{corner}}/f_{\text{signal}} > 30$ if using input cell filter
- **Lock-in detection averages out noise & signals except at harmonics of the modulation frequency**
 - Anti-aliasing not necessary (but will improve SNR if used)

Temperature-sensing (1)

○ Thermistors

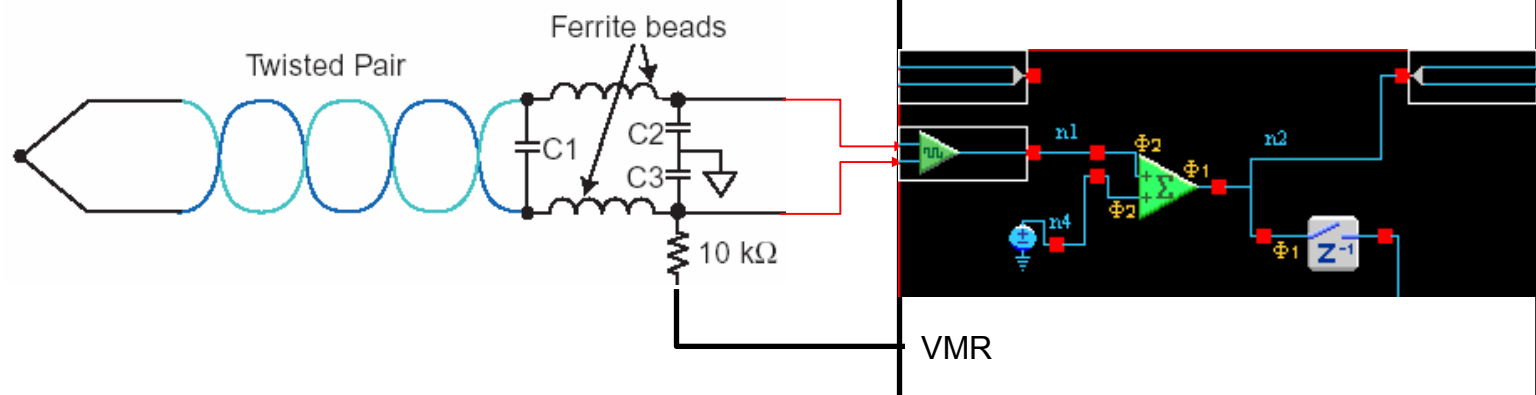
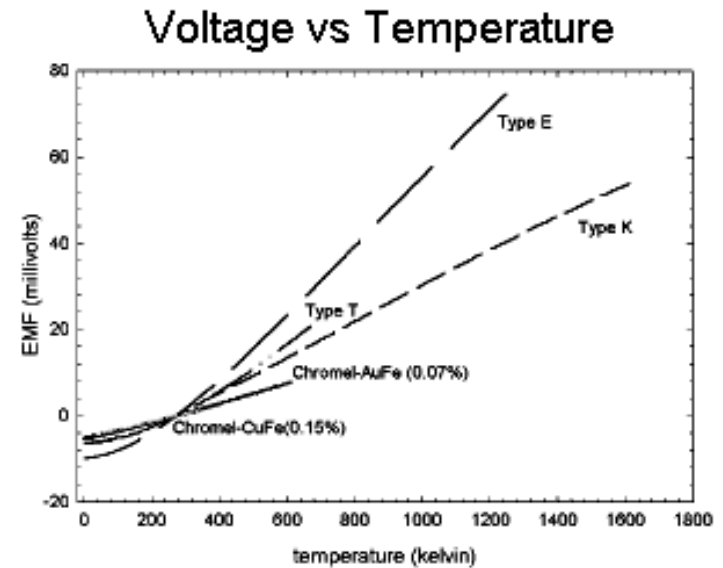
- Resistances range from 100 to 1Mohm
- Typical current < 100uA to avoid self- heating



Temperature-sensing (2)

- **Thermocouples**

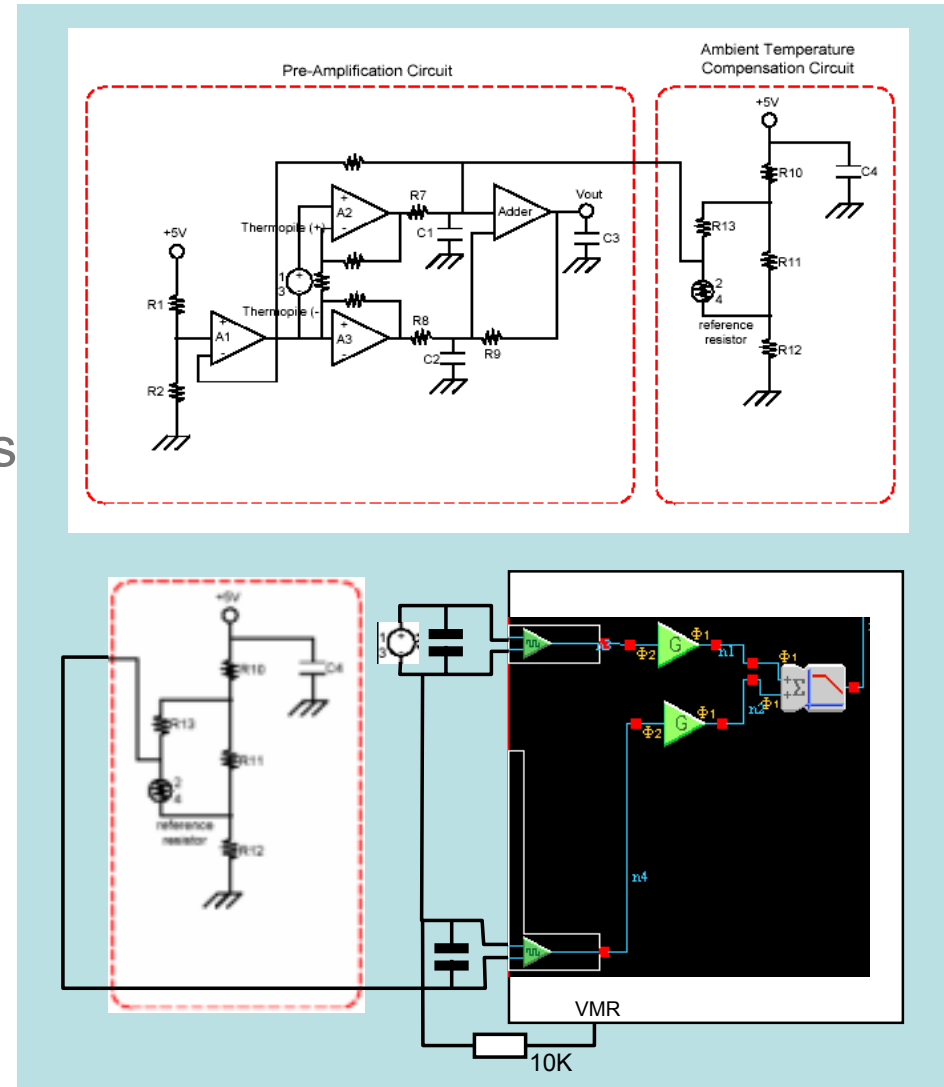
- Small voltage/degree C
- Prone to high levels of CM noise



Temperature-sensing (3)

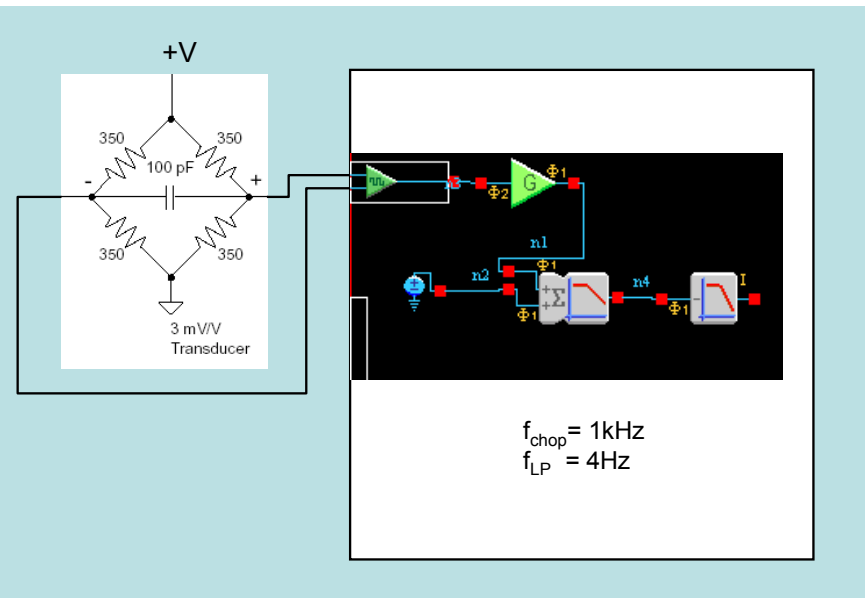
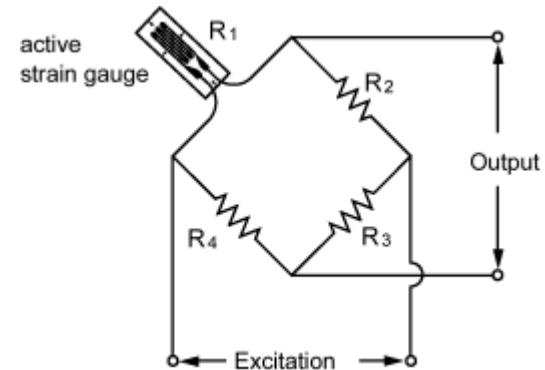
- **Thermopiles**

- Serially-interconnected thermocouples with “hot” and “cold” junctions
- “Hot” junction sensitive to received IR radiation
- Close proximity of junctions gives low sensitivity to ambient temperature



• Strain gauges

- Typical output < 10 mV/V
- “Half bridge” has strain gauges in two arms:
 - doubles the output and compensates for thermal effects
- “Full bridge” has strain gauges in four arms:
 - re-doubles output and compensates for thermal effects.



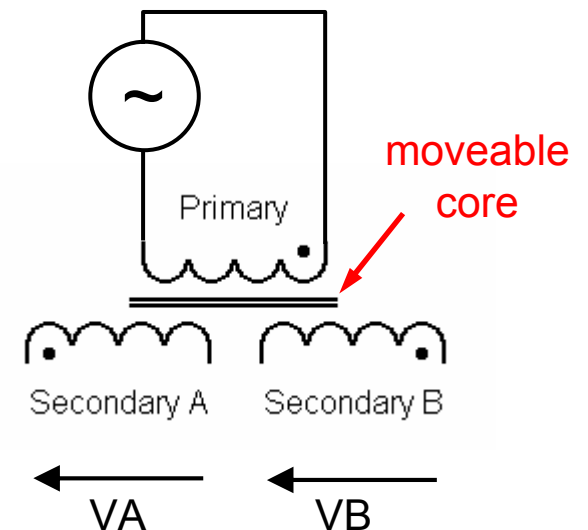
● LVDT

• Sensitivity ranges:

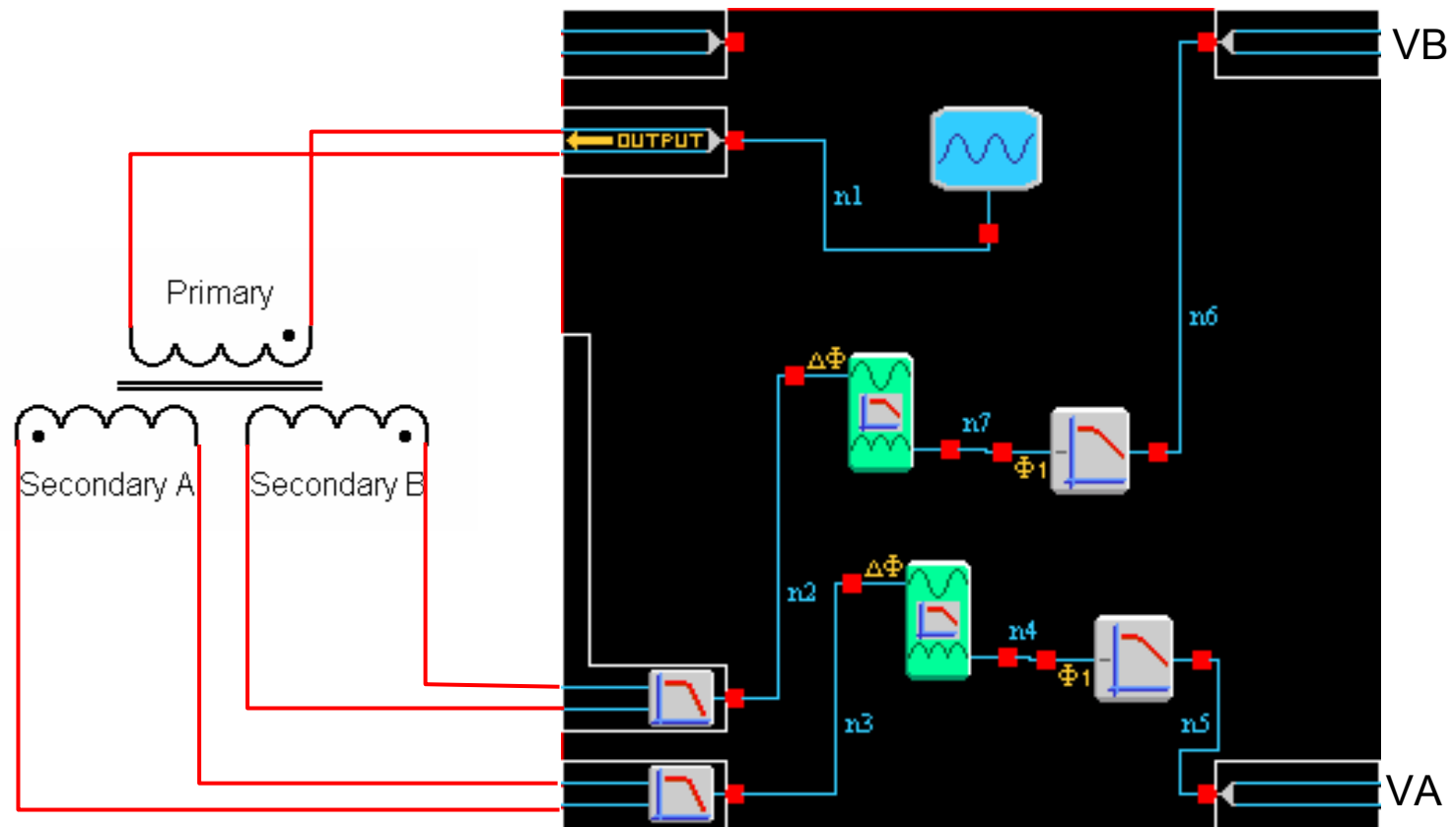
- 0.05 mV/V/0.001" for long stroke LVDTs
- 10 mV/V/0.001" for short stroke LVDTs

• Various signal processing techniques (eg):

- Rectify and filter AC signals, then:
 - Calculate $(V_A + V_B)/V_{\text{primary}}$ (insensitive to the amplitude of the driving signal, and gives some noise rejection).
 - Calculate the **ratio** of the difference and the sum of the secondary voltages i.e. $(V_A - V_B) / (V_A + V_B)$
- Measure the phase of the combined $V_A + V_B$ secondary voltage i.e. -180 degrees at one extreme, zero degrees at the null position, and +180 degrees at the other extreme.



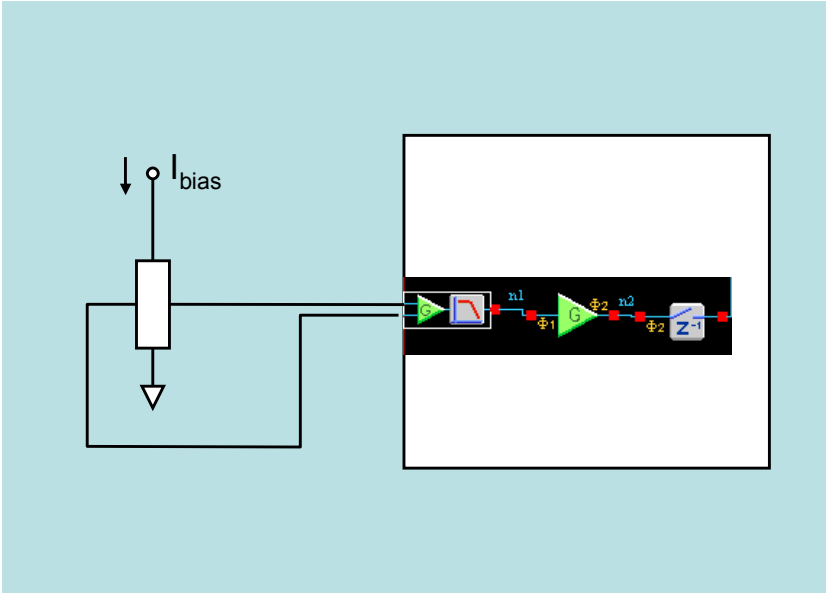
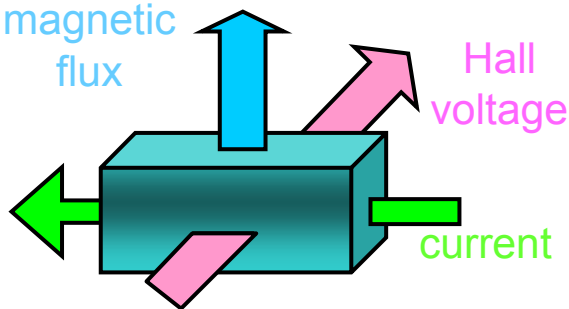
LVDT sensing



Magnetic-sensing

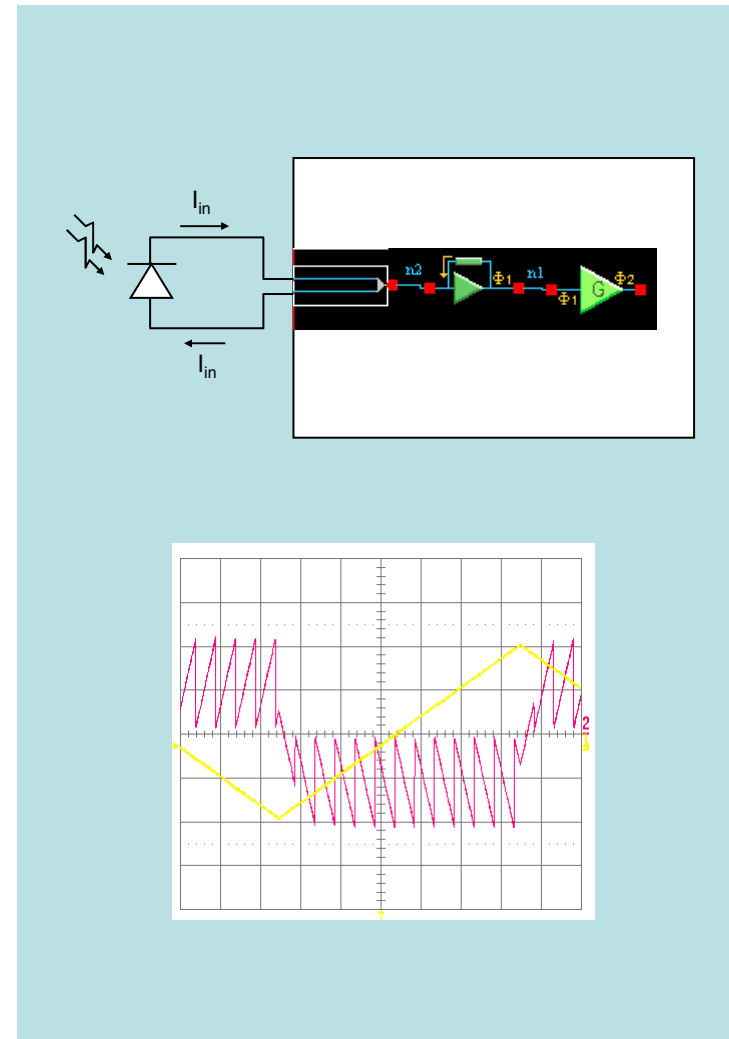
- **Hall effect devices**

- Usually driven with a **constant current**
- Differential output voltage, superimposed on a common-mode voltage approximately equal to half the excitation voltage.
- Typical sensitivity:
 - 1-100 mV/kG
 - (Refrigerator magnet: 200 gauss)
- Typical element resistance:
 - 1 to 10 ohms
- Typical excitation current:
 - 20 to 200 mA.
- Typical linearity:
 - 0.1% to 2%.

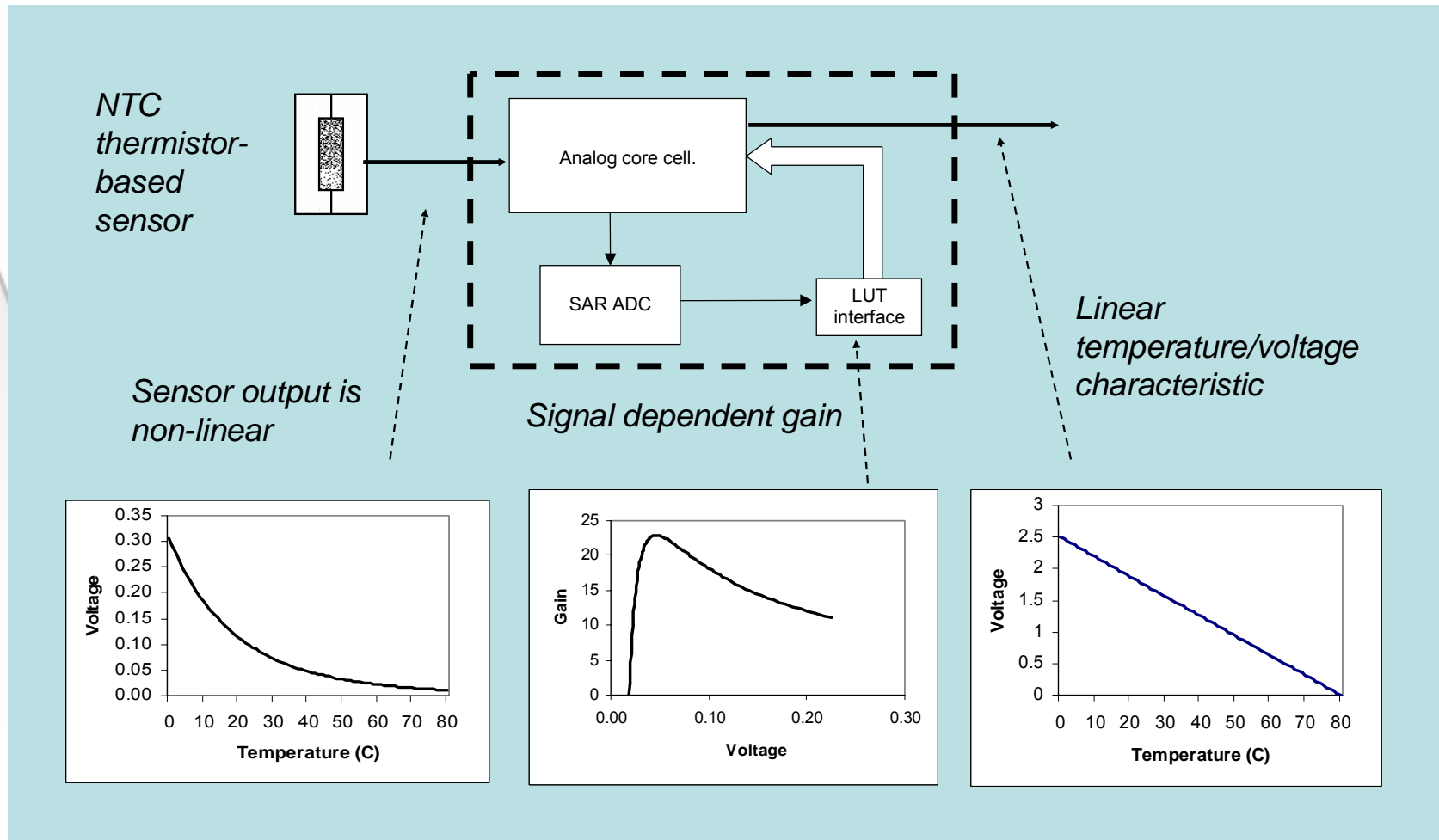


- **Photodiode/transistor**

- Photo-generated current develops a voltage across a feedback “resistor”
- Transimpedance is dependent on absolute capacitor values, **not** a capacitor ratio
- Transimpedance CAM output is only valid at the end of each clock phase – follow by an appropriate CAM

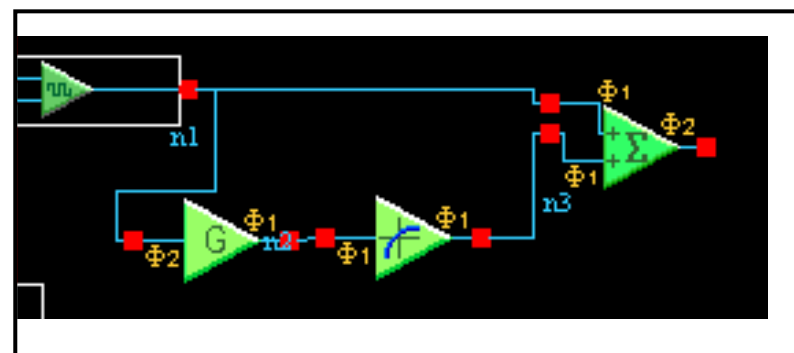
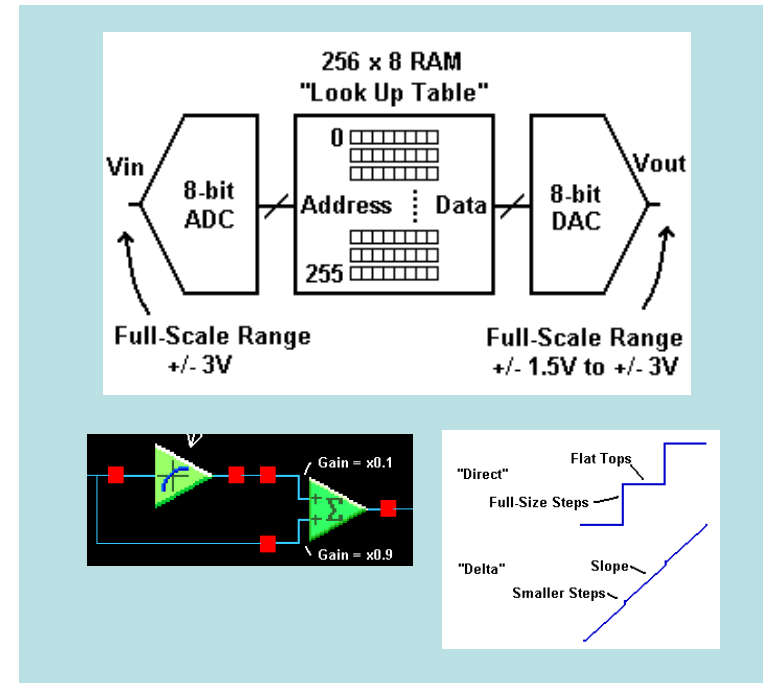


Sensor linearisation



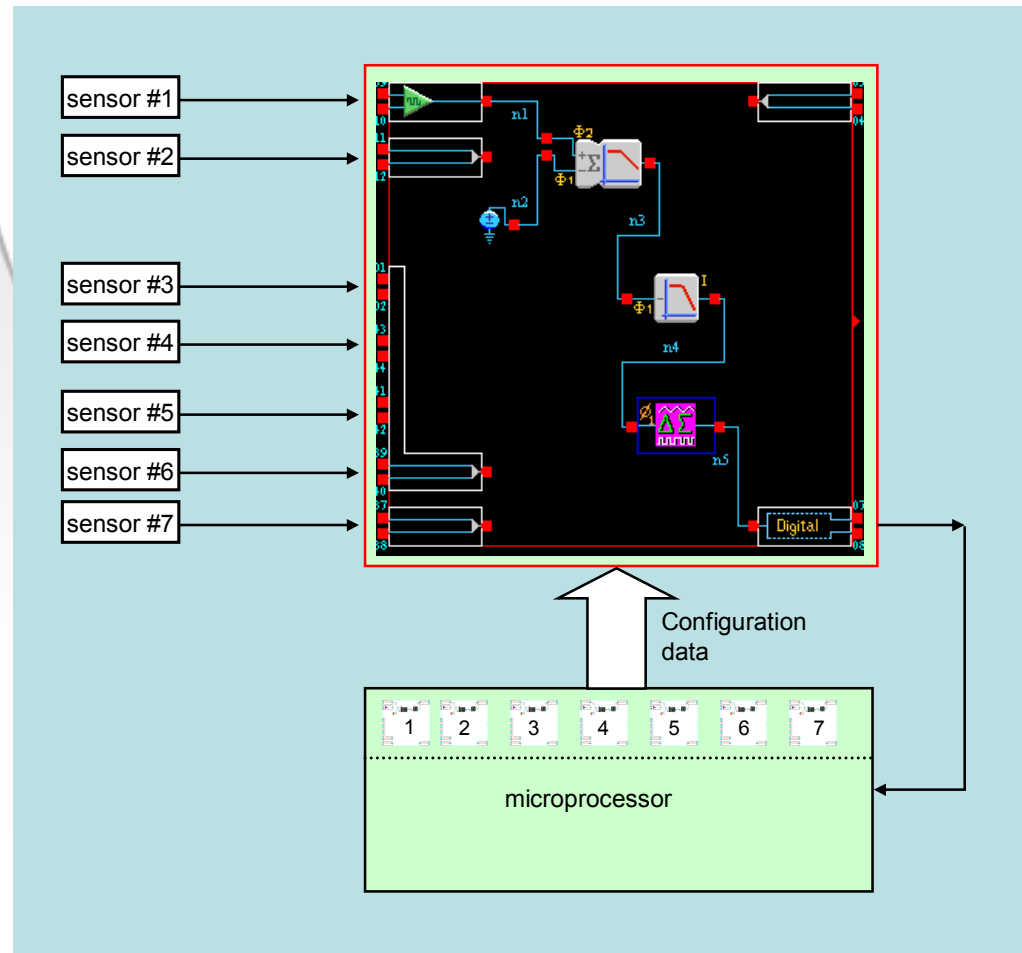
- **TransferFunction
CAM**

- Quantises gain – use full-scale input & output for minimum non-linearity
- When the input signal is nearly linear, use “delta” information rather than “direct”



Hardware multiplexing

- State-driven dynamic reconfiguration:



```
void main()
{
    // Get the Primary Configuration data
    int dataSize = 0;
    const an_Byte* pData = an_GetPrimaryConfigData(an_chip1_Primary, &dataSize);

    // Send it
    Download(pData, dataSize, 1);

    an_Circuit nCircuit = an_circuit1_chip1_Data;
    while (1)
    {
        // Work out which circuit is needed next
        switch(nCircuit)
        {
            case an_circuit1_chip1_Data:
                nCircuit=an_circuit2_chip1_Data;
                break;
            case an_circuit2_chip1_Data:
                nCircuit=an_circuit3_chip1_Data;
                break;
            // etc., for the other cases
            default:
                nCircuit=an_circuit1_chip1_Data;
        }

        // Get the transition data
        dataSize = 0;
        pData = an_GetCircuitTransitionData(nCircuit, &dataSize);

        // Send it
        Download(pData, dataSize, 0);

        // Pause while the measurement is taken
    }
} //end of main()
```

- **Guidelines for sensor/measurement signal conditioning:**
 - Select suitable clock(s) and watch out for signal aliasing effects
 - Use appropriate offset removal & compensation techniques
 - Think about output waveforms and smoothing
- **Systems may benefit from dynamic configuration and hardware multiplexing**