The FPAA Company

Field Programmable Analog Arrays

real time Analog programmability

2016 Technical Training, Class IV

anadigm™

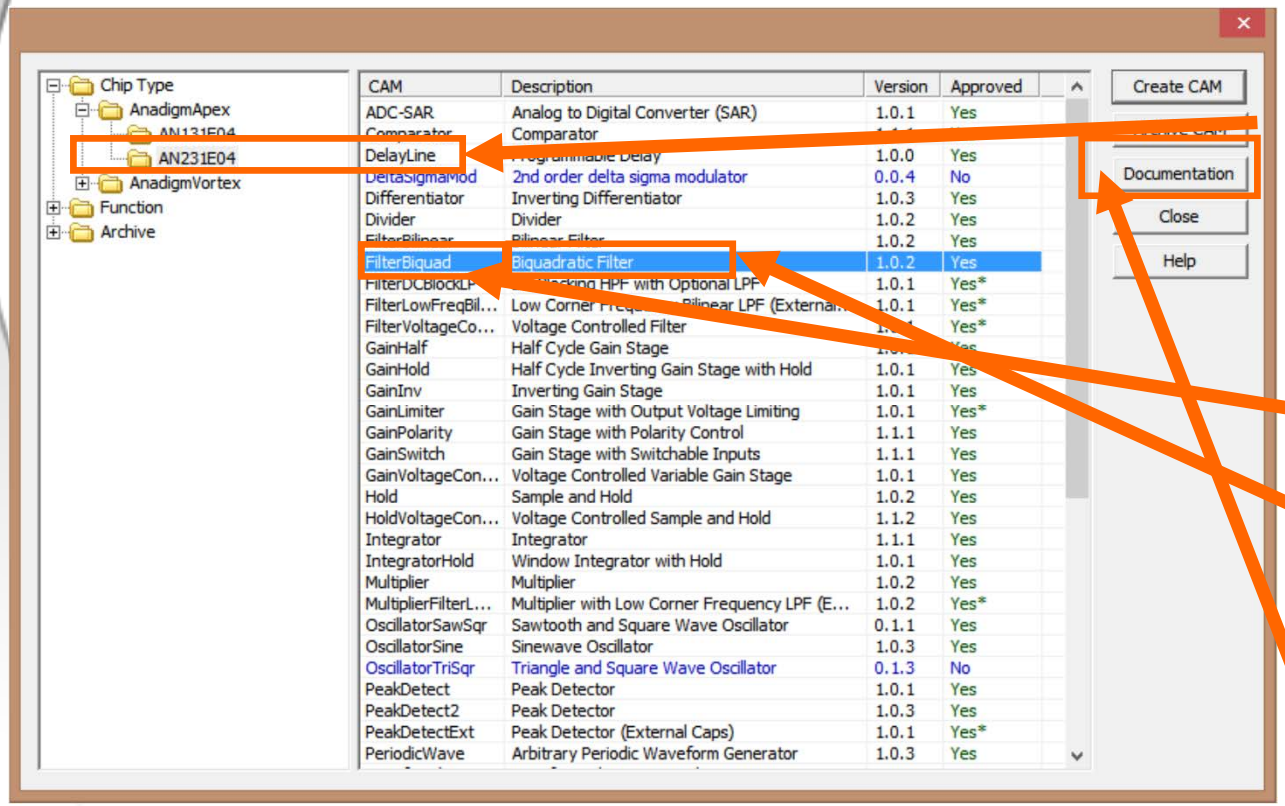
What is a Configurable Analog Module (CAM)?

- **Circuit building blocks abstracted to a functional level that can be manipulated in AnadigmDesigner®2**
- **A complex circuit can be implemented in a "chip" simply by selecting, configuring, placing and wiring CAMs**
- **Improved speed and ease of circuit design**

AnadigmDesigner®2 CAMs

- **Very dynamic – powerful yet easy to use**
  - Multiple circuit topologies – CAM knows how to make what you ask for
  - Dynamic user interface – options and limits can change
    - Allows user to push the limits of the CAM
    - Constrains the user to legal configurations
- **Expanded CAM documentation explains the features**

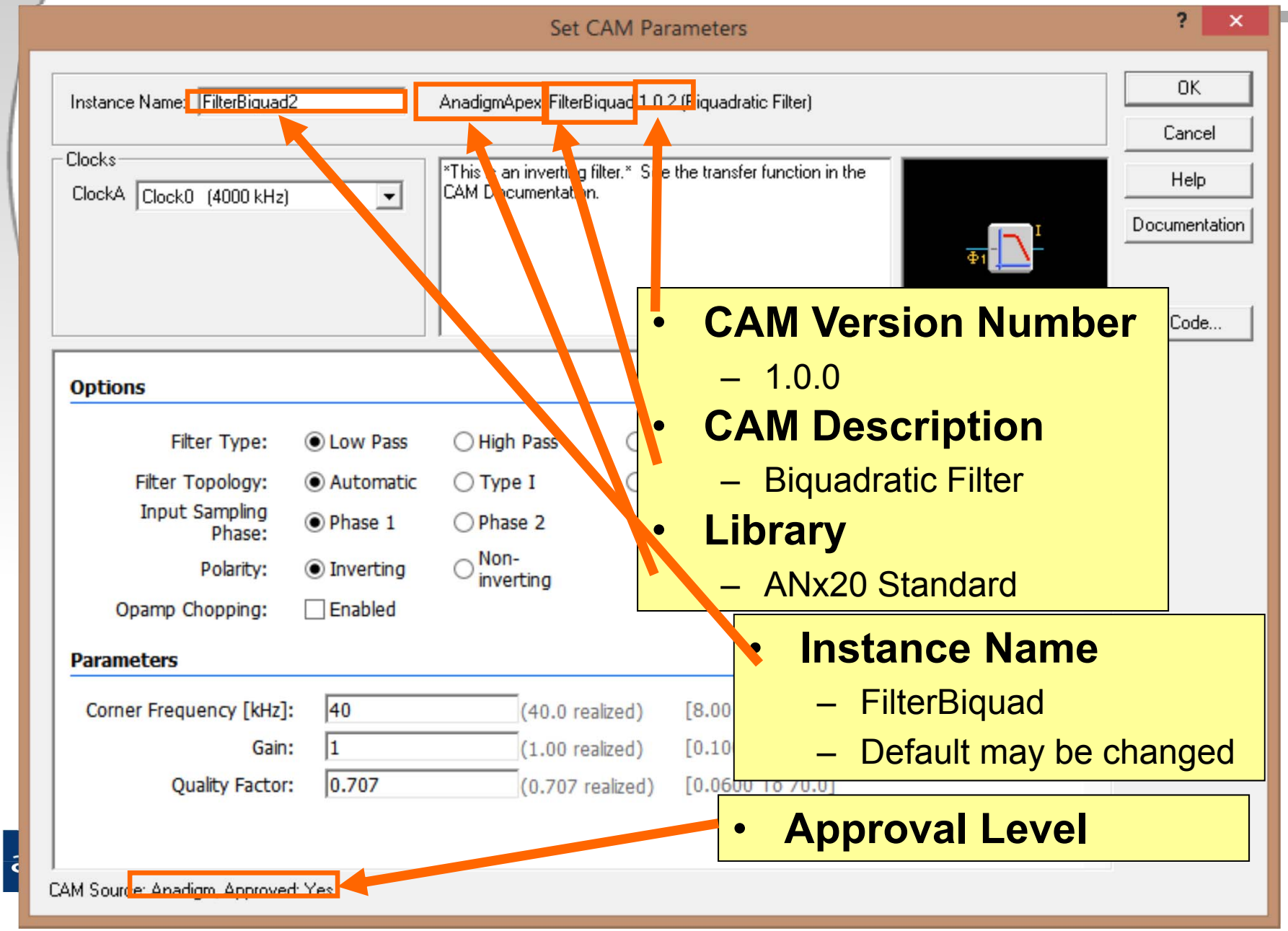
Selecting a CAM



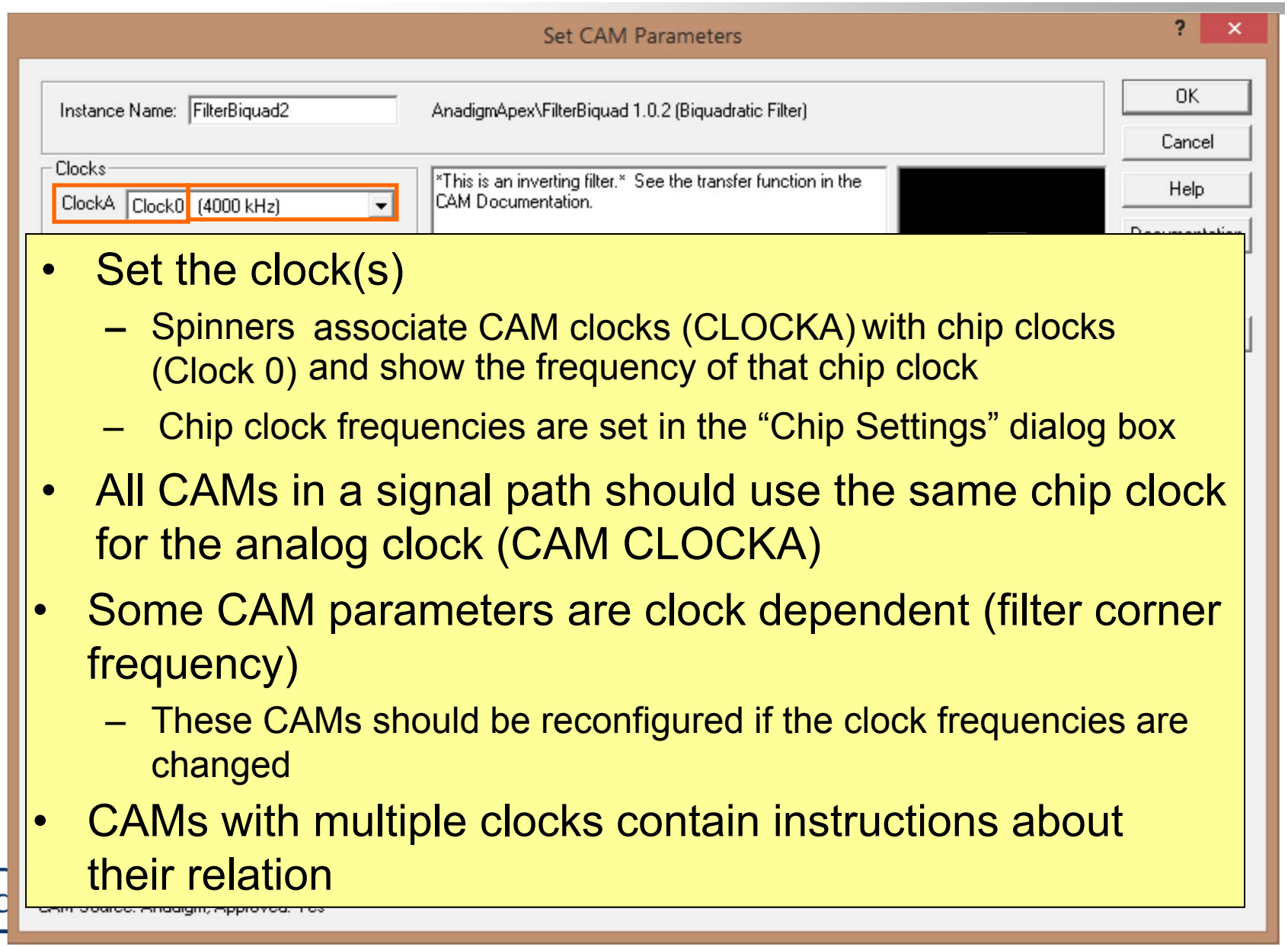
- **Library**
  - **AnadigmApex**
  - **AN231E04**
- **Name**
  - **FilterBiquad**
- **Description**
  - **Biquadratic Filter**
- **Documentation**

In AnadigmDesigner®2, CAMs may contain multiple circuit configurations. Select the basic function. The details will be set during CAM configuration.

Configuring the CAM - Information

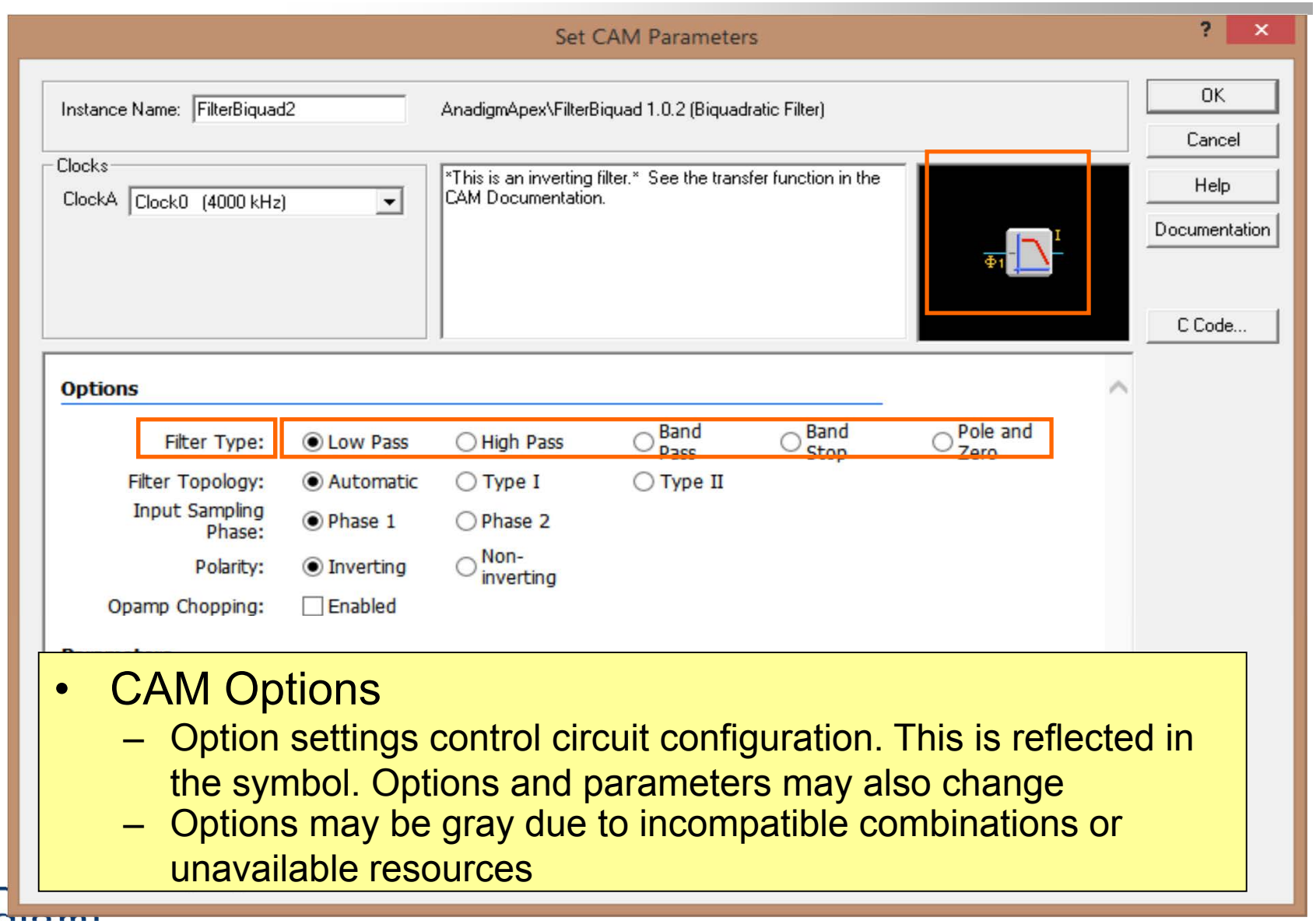


Configuring the CAM - Clocks



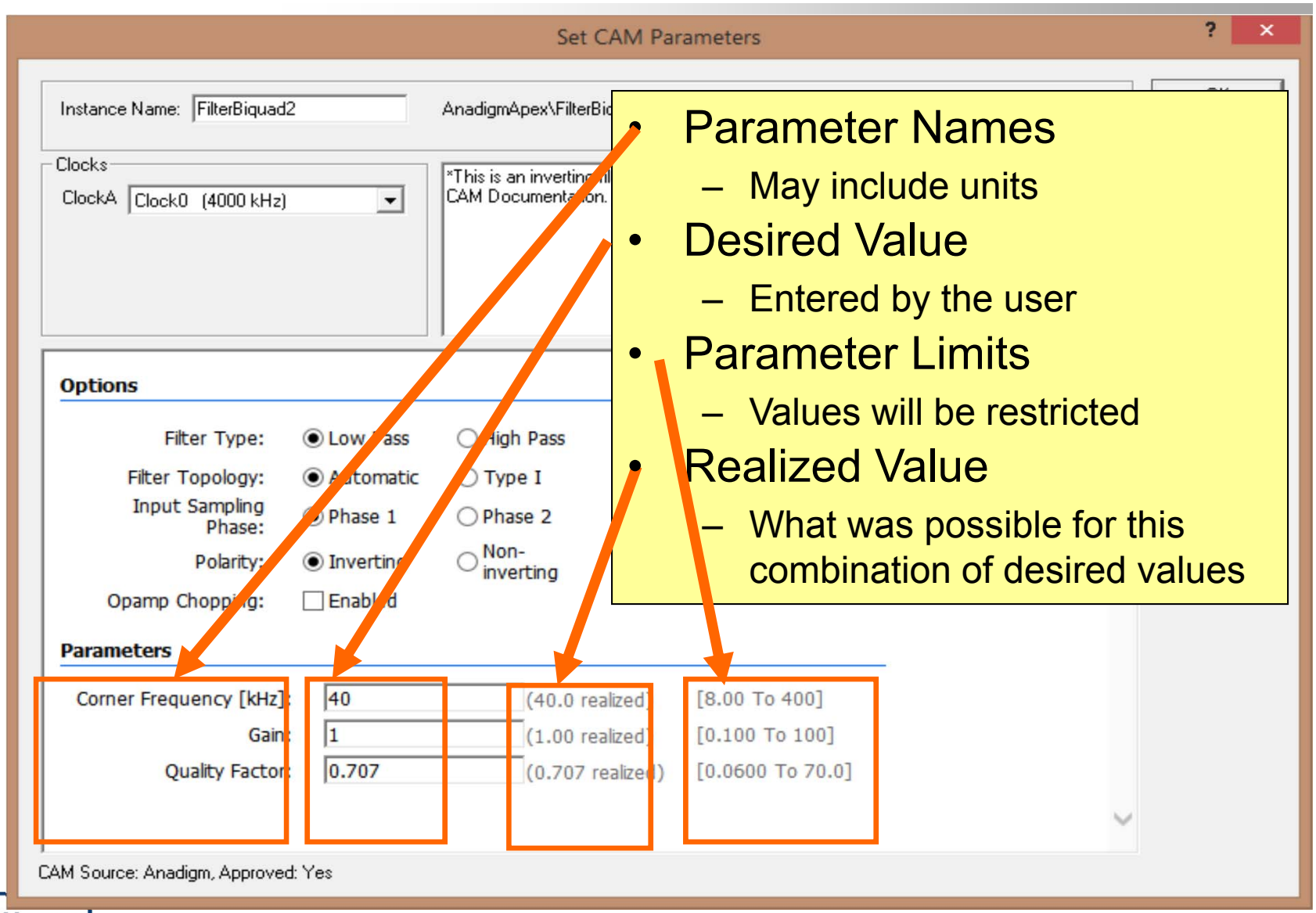
- Set the clock(s)
  - Spinners associate CAM clocks (CLOCKA) with chip clocks (Clock 0) and show the frequency of that chip clock
  - Chip clock frequencies are set in the “Chip Settings” dialog box
- All CAMs in a signal path should use the same chip clock for the analog clock (CAM CLOCKA)
- Some CAM parameters are clock dependent (filter corner frequency)
  - These CAMs should be reconfigured if the clock frequencies are changed
- CAMs with multiple clocks contain instructions about their relation

Configuring the CAM - Options



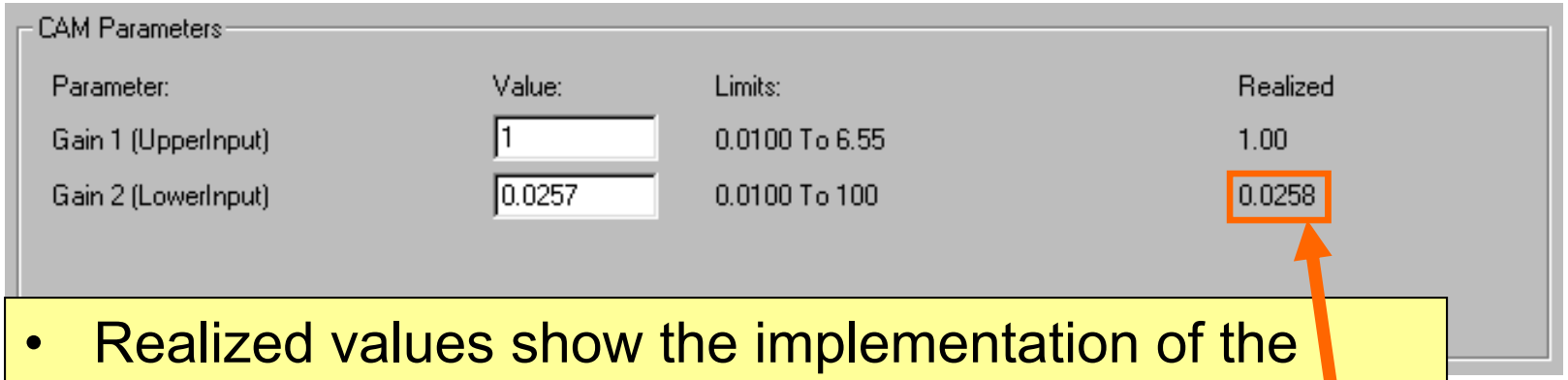
- CAM Options
  - Option settings control circuit configuration. This is reflected in the symbol. Options and parameters may also change
  - Options may be gray due to incompatible combinations or unavailable resources

Configuring the CAM - Parameters



- Parameter Names
  - May include units
- Desired Value
  - Entered by the user
- Parameter Limits
  - Values will be restricted
- Realized Value
  - What was possible for this combination of desired values

Parameters – Quantization and Error



- Realized values show the implementation of the parameter based on ratios of programmable capacitor banks which are **quantized**

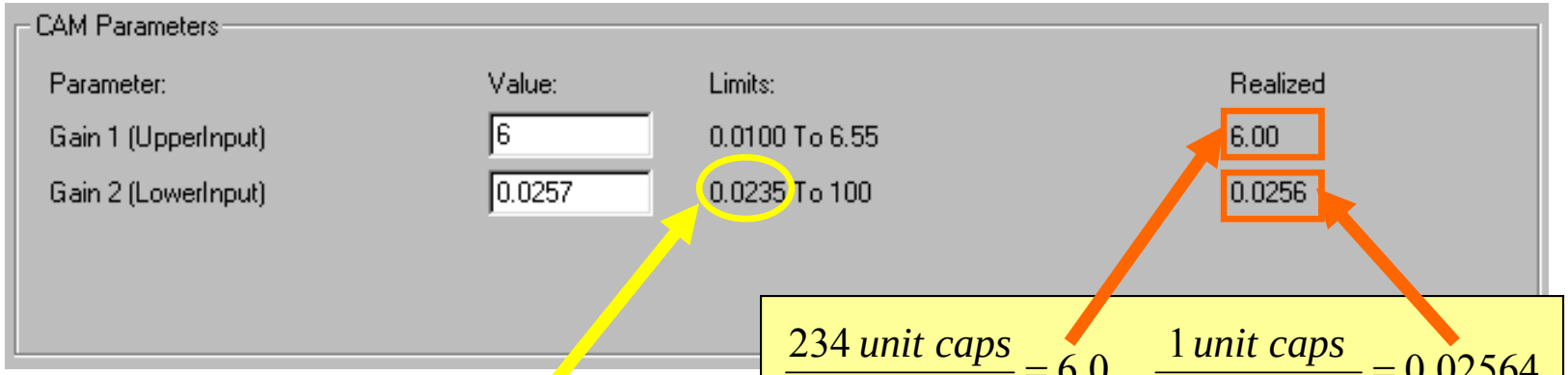
$$\frac{6\ unit\ caps}{233\ unit\ caps} = 0.02575$$

- Actual measured values can have **error** in addition to the quantization of the realized value

$$Gain_{Realized} = 0.02575$$

$$Gain_{Measured} = 0.0259 \Rightarrow 0.6\ \%\ error$$

Parameters - Interrelation



$$\frac{234\ unit\ caps}{39\ unit\ caps} = 6.0 \qquad \frac{1\ unit\ caps}{39\ unit\ caps} = 0.02564$$

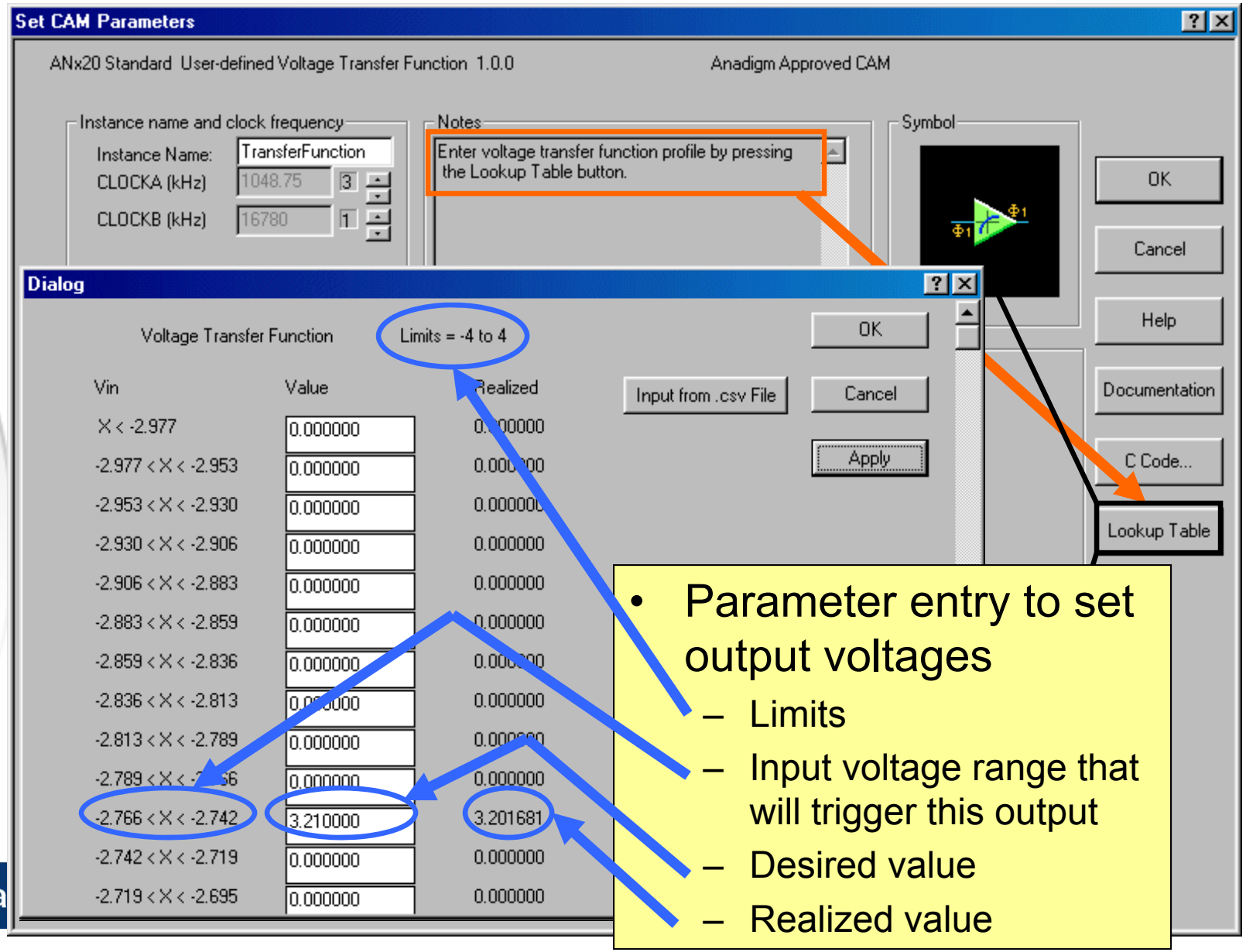
- Realized values are based on the combination of capacitor ratios. Changing one desired value can change multiple realized values.

- Limits are dynamic. Changing desired values can also change the limits.
  - If Gain 1 = 6.0

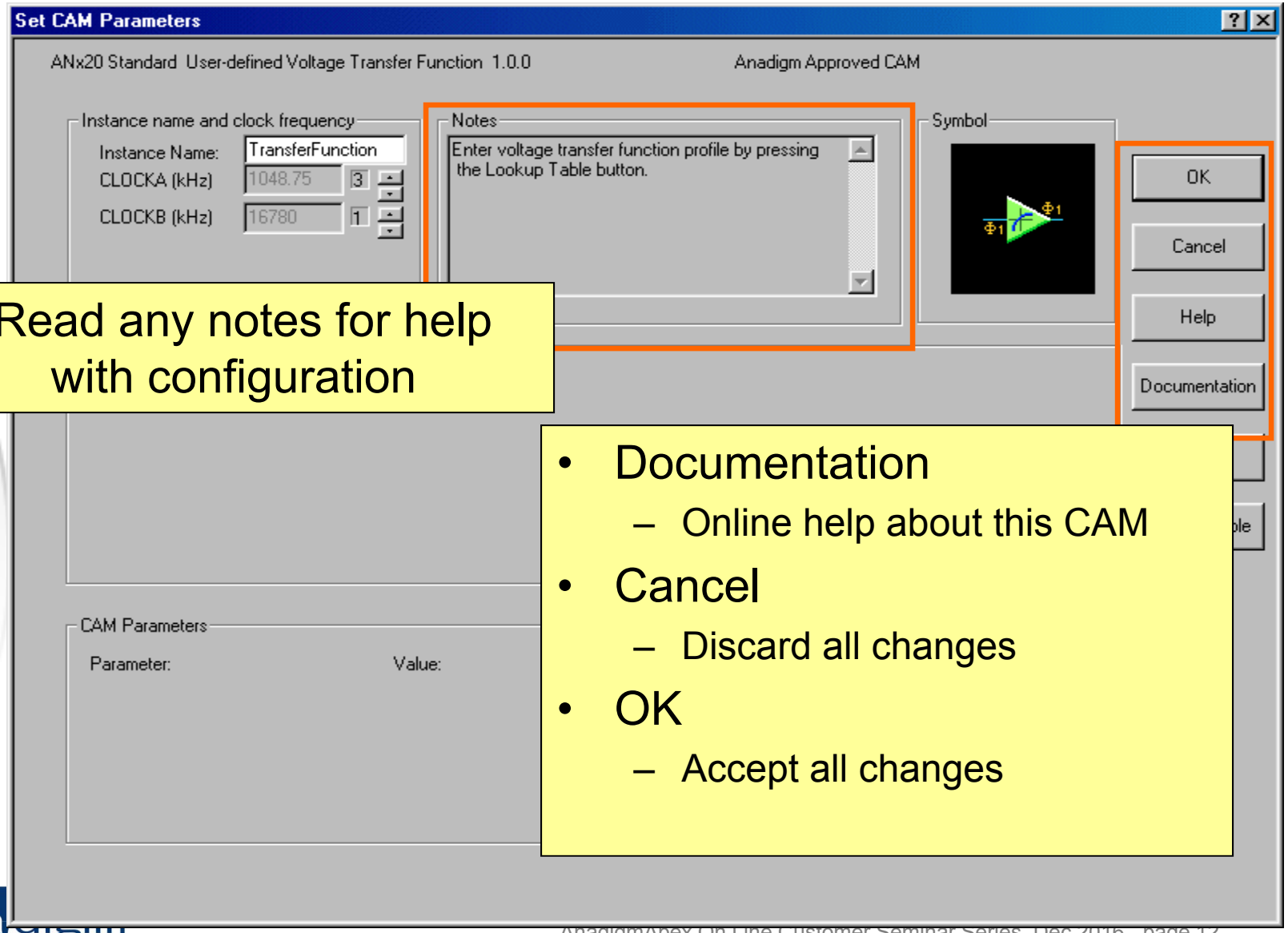
  Gain 2 cannot be less than 0.0235
  - If Gain 2 = .0257

  Gain 1 cannot be greater than 6.55

Configuring the CAM - LUT



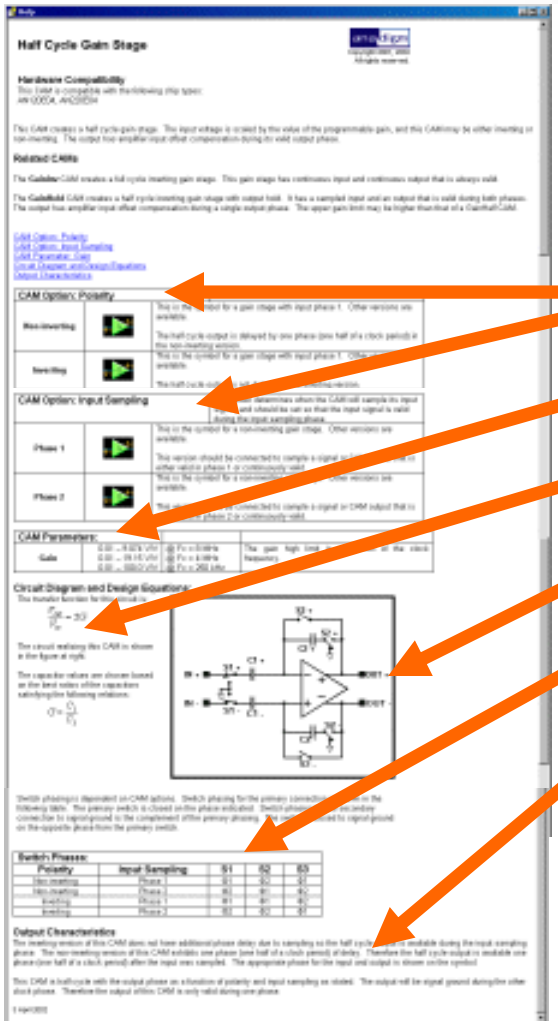
Configuring the CAM - Finishing



Read any notes for help with configuration

- Documentation
  - Online help about this CAM
- Cancel
  - Discard all changes
- OK
  - Accept all changes

Online CAM Documentation

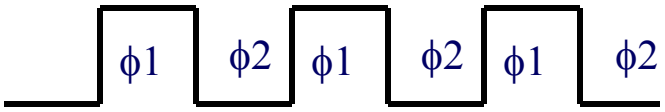


- Anadigm approved CAMs contain information about CAM construction and proper usage
  - Details about each CAM option
  - Details about each CAM parameter
  - Design Equations
  - Circuit Diagrams
  - Switch Phasing
  - Output Characteristics
- Some include additional design notes with information about special features of that CAM

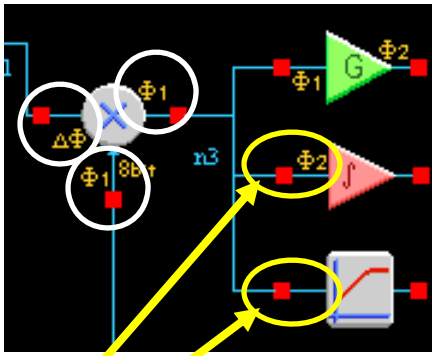
Placing and Wiring CAMs

- **Place the CAM within the chip borders**
  - Green warning marker indicates the CAM cannot be dropped on top of something
  - Red warning marker indicates that available resources are not sufficient to implement the CAM
- **Draw wires between the CAM contacts**
  - Only legal connections will be allowed
- **Chips can be connected for simulation**

Clock Phases



- Each clock has two non-overlapping phases
- Phase symbol on a CAM input shows an input that samples only on that phase
  - $\Delta\phi$ indicates that the sampling phase changes during operation
- Phase symbol on a CAM output shows the output should be sampled on that phase

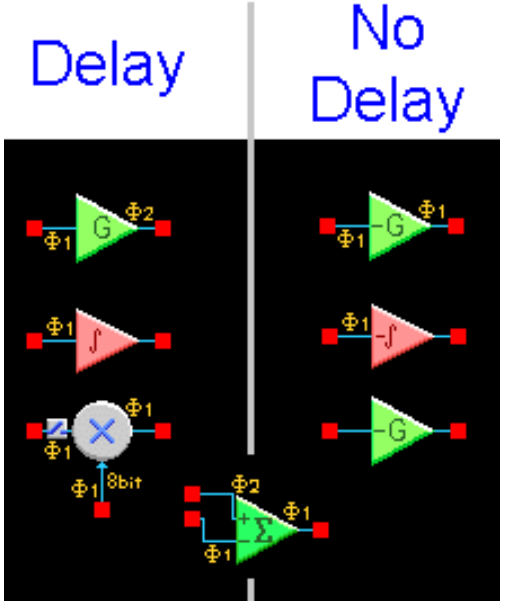


- **Warning:** a phased output can be safely connected only to a similarly phased input
- Always see the CAM documentation for details on input/output characteristics

Clock Delay

- CAMs may have signal delay due to the timing of clocked switches. This is not the same as filter phase delay.
- Clock delay can often be neglected if the clock frequency is adequately higher than the signal frequency

Example – 10 kHz signal CAM has half clock cycle delay
  – With 50 kHz clock
  36 degree delay (possibly significant)
  – With 1 MHz clock
  1.8 degree delay (probably negligible)



- Clock delay is not shown by symbol alone
- Always see the CAM documentation for details on input/output characteristics

CAM Files

.cam File

- **Primary CAM file**
- **ASCII based**
- **Read directly by AnadigmDesigner®2**
- **Strictly formatted, keyword driven with very little error checking**

  Name, Version, User Interface Control, Circuit Definition, Parameter Calculation, Symbol, Simulation equations, CCODE, etc.

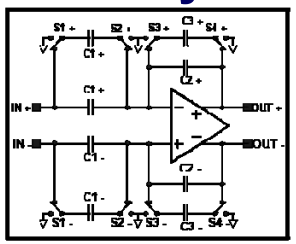
.chm File

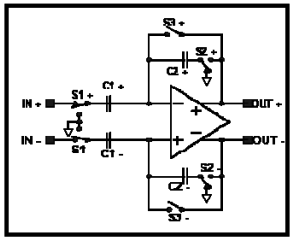
- **CAM Documentation or Help file**
- **Compiled HTML**
- **Referenced and displayed by AnadigmDesigner®2**

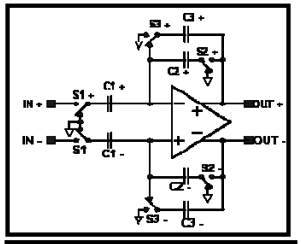
CAM Gain Elements

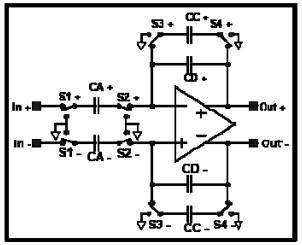
There are four basic gain topologies that are reused in many CAMs (gain stage, rectifier, summing stage, etc)

- **xxxInv**
  - Inverting
  - Continuous time – the input is not sampled
- **xxxHalf**
  - Inverting or non-inverting
  - Amplifier input offset compensation
  - Half-cycle (Output is zero during one phase)
  - Subject to clock frequency/gain limitations
- **xxxHold**
  - Inverting
  - Amplifier input offset compensation for only one phase
- **xxxFilter – uses a single pole low pass filter**
  - Inverting or non-inverting









ANx20 Standard Library – Gain Stages

- **GainHalf** 

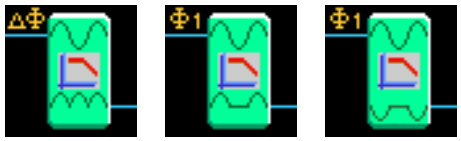
  - Half-cycle
- **GainHold** 

  - Inverting only
- **GainInv** 

  - Continuous Time

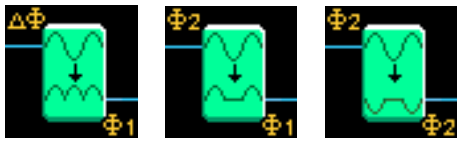
ANx20 Standard Library – Rectifiers

- **RectifierFilter**



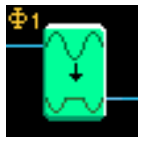
  - Full Wave/Half Wave
  - Inverting/non-inverting

- **RectifierHalf**



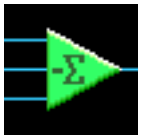
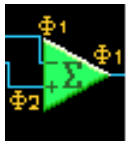
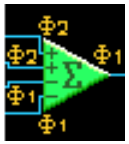
  - Full Wave/Half Wave
  - Inverting/non-inverting

- **RectifierHold**

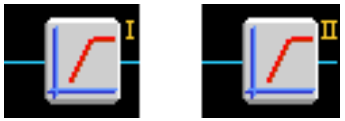


  - Half Wave Inverting only

ANx20 Standard Library – Summing

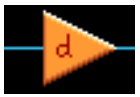
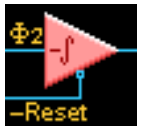
- **SumInv** 
  - Up to three inputs
- **SumDiff (SumHalf)**  
  - Up to four inputs
  - Add or subtract since input branches can be inverting or non-inverting

ANx20 Standard Library – Filters

- **FilterBilinear – One pole** 
  - Low Pass/High Pass/All Pass
- **FilterBiquad – Two poles** 
  - Low Pass/High Pass/Band Pass/Band Stop
  - Automatically chooses from multiple circuit topologies 

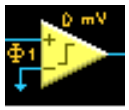
Some other CAMs use a low pass bilinear filter as part of another function (RectifierFilter)

ANx20 Standard Library – Math

- **Differentiator** 
  - Output voltage slews – see documentation
- **Integrator** 
  - Optional reset 

ANx20 Standard Library

- **Comparator**

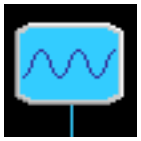


  - Single/Dual Input

  - Variable Reference

- **Hold – Sample and hold**

- **OscillatorSine**



  - Subject to internal reference voltage error

- **Voltage (+/- 3 VDC)**

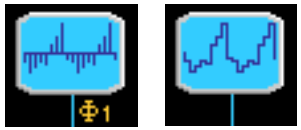
  - Subject to internal reference voltage error

ANx20 Standard Library – Multiplier

- **Multiplier** 

  - Uses SAR (Input Y is quantized)
  - Subject to internal reference voltage error
  - Optional sample and hold on input X to equalize sampling time of two inputs (uses chip resources)

ANx20 Standard Library – LUT

- **PeriodicWave**
  - Half-cycle/Output Hold 
  - Uses LUT to generate a user-defined periodic sequence of output voltages
  - Documentation has help with loading the LUT
- **TransferFunction**
  - Half-cycle/Output Hold 
  - Uses the SAR and LUT to perform A/D conversion on the input and generate the appropriate user-defined output voltage