

## Major components of the Synchro DSP

Function - Create floating point sine, cosine lookup tables

**PRINCIPAL ROUTINE - c\_int26()**  
I SR running at 50kHz clocked by the Pentek 6102 ADC.

1. read input comm-port A3. Decode 32 bit word into 16 bit X and Y data.
- 2 (i) if 1 in 50 countdown (for 1kHz loop) {
3. (i) if phase rampdown flag set {
3. (ii) apply successive small-step decrement to rampdown factor.
3. (iii) compute beam phase = initial reference phase \* decreasing rampdown factor
3. (iv) reset flags when done }
4. (i) if cogging flag set {
4. (ii) apply successive small-step cogs with chosen +ve or -ve slope.
4. (iii) modulate phase between -PI and PI
4. (iv) reset flags when done }
5. compute sine and cosine of beam phase via lookup table for efficiency.
6. end 1 in 50 countdown (for 1kHz loop)
7. implement synchro rotation  $Y'(t) = Y \cos(\phi) - X \sin(\phi)$ .
8. send  $Y'(t)$  to synchro feedback loops via shared DRAM.

Trigger Receive ISR - c\_int30() - triggered by an internal comm port write from I/O DSP

1. read comm-port A4. Decode the trigger index.
2. If trigger index = Synchro Start {
3. (i) calculate phase reference on current X, Y data. ( $\text{atan2}(Y/X)$ )
3. (ii) implement synchro rotation  $Y'(t) = Y \cos(\phi) - X \sin(\phi)$ .
3. (iii) chain trigger to waiting feedback loops.
3. (iv) enable rampdown }
4. If trigger index = Cogging Start {
5. (i) read integer buckets to cog from VSB bucket counter.
5. (ii) enable  $N * 2\pi$  cogging }

Forever loop - executed at lower priority than ISRs above

1. output  $\phi(t)$  and  $Y'(t)$  to DAC via comm-port A0
2. if Update button pressed {
3. (i) read new user inputs
3. (ii) if synchro in use ie synchro start and rampdown completed.
3. (iii) enable/perform manual cog tweak (+ve, -ve, fractional)