Understanding and Working With ARM Bit-Packed QC

Tim Shippert
Bit-Packed QC

• New VAPs are including bit-packed qc values following the ARM qc standards
• Little info available to ARM users/developers on how to work with bit-packed QC
• Tim has put together a tutorial with examples in several languages:
  https://engineering.arm.gov/~shippert/ARM_bits2.html

Comments/suggestions/additional examples welcome
Introduction to bit-packing

• ARM qc tests are binary – data either passes or fails each test
• Each bit contains information about a particular QC test and has a value of either “0” – test passed or “1” – test failed
• The final QC value is the sum of the bits that failed the tests:

\[ QC = \sum_{b=1}^{32} t(b)2^{b-1} \]

where \( b \) is the number of the test, or bit, and \( t(b) \) is the result (0 or 1) of each test

• To test if a particular bit is set, bitwise AND the QC value with the integer corresponding to that "clean" bit - i.e. the integer that has that bit set to "1" and all other bits set to "0"
• For a clean bit representation of bit N, shift the integer "1" left by N-1 positions: \( \text{bit5} = \text{left\_shift}(1, 5-1) \)
• Test if bit5 is set: if \(((\text{qc\_field AND bit5}) \text{ NE 0}) \{ \text{print “Bit 5 is set in qc\_field” } \} \)
FUNCTION build_mask, bits
    mask=0UL
    FOR i=0, n_elements(bits)-1 DO BEGIN
        mask = (mask OR ishft(1UL, bits[i]-1)) ENDFOR
    END

x=[1,2,4,7]
mask=build_mask(x)
if ((qc_val AND mask) ne 0) then begin
    print, "Qc_val failed tests 1, 2, 4, and/or 7!"
endif
Additional Topics in Tutorial

• Reading multiple QC tests with masks
• Writing qc tests as bit-packed integers
• Signed vs unsigned integers
• Clearing bits
• Examples in C, Fortran, IDL
• Upcoming: examples in Matlab (Connor) and Python (Justin)