Algorithm-based Fault Tolerance (ABFT)

♦ Data redundancy at the application level
  * Higher efficiency when applied to large data arrays
  * Examples: matrix-based and signal processing applications
  * Given \( n \times m \) matrix \( A \) define the column checksum matrix
    \[
    A_C = \begin{bmatrix} A \\ eA \end{bmatrix}
    \]
    where \( e = [1 \ldots 1] \)
  * Row check matrix \( A_R = [A Af] \) where \( f = [1 \ldots 1]^T \)
  * \((n+1) \times (m+1)\) full checksum matrix
    \[
    A_F = \begin{bmatrix} A Af \\ eA eAf \end{bmatrix}
    \]
  * Column and row checksum matrices can detect a single fault
  * Full checksum matrix can locate a fault – if checksum accurate the fault can be corrected
ABFT for Matrix Operations

♦ Matrix addition $A+B=C$ can be replaced by
  $A_C + B_C = C_C \text{ or } A_R + B_R = C_R \text{ or } A_F + B_F = C_F$

♦ Matrix multiplication $A \times B = C$
  $A_B = C_R \text{ or } A_C B = C_C \text{ or } A_C B_R = C_F$

♦ Faults can be located and corrected by adding weighted checksum row(s) or column(s)

\[
A = \begin{bmatrix} A & eA \\ e_A & A_w \end{bmatrix} \text{ where } e_w = [1,2 \cdots 2^{n-1}] \\
A_R = \begin{bmatrix} A & Af_w \\ e_A & Af_w \end{bmatrix} \text{ where } f_w = [1,2 \cdots 2^{m-1}]^T \\
A_F = \begin{bmatrix} A & Af_w \\ e_A & Af_w \\ e_A & Af_w \end{bmatrix}
\]

Weighted Checksum Code (WCC)

♦ Example for single error correction: $A_C = \begin{bmatrix} A \\ eA \\ e_A \end{bmatrix}$
  * Suppose an error detected in column $j$
  * WCS1/WCS2 unweighted/weighted checksum $eA/e_A$ for column $j$
  * Calculate error syndromes:
    \[
    S_1 = \sum_{i=1}^{n} a_{i,j} - WCS1 \\
    S_2 = \sum_{i=1}^{n} 2^{i-1} a_{i,j} - WCS2
    \]
  * If only one syndrome is nonzero - the checksum is wrong
  * If both are nonzero $S_2/S_1 = 2^{k-1}$ implying that $a_{k,j}$ is in error
    \[
    a_{k,j} = a_{k,j} - S_1
    \]
  * Extra rows (columns) can be added with
    \[
    e_w = [1 \cdots 2^{d-1} \cdots (2^{n-1}) \cdots 2^{d-1}] \\
    f_w = [1^{d-1} 2^{d-1} \cdots (2^{m-1}) \cdots d-1]^T
    \]
  * The weighted checksum with $v$ rows - Hamming distance $v+1$: detecting $v$ or correcting $\lfloor v/2 \rfloor$ errors
Checksum Overflow

♦ For large $n$ and $m$ checksums can overflow
  * Single-precision 1-bit checksum - result calculated mod-$2^l$
  * A single error < $2^l$ and will be detected
  * Weighted checksum would need more bits to avoid overflow
  * Instead of $e_w = [1 \ 2 \ \cdots \ 2^{n-1}]$ use $e_w = [1 \ 2 \ \cdots \ n]$
  * If both syndromes for column $j$ are nonzero and $S_2/S_1 = k$ 
    $$a_{k,j} = a_{k,j} - S_1$$
  * Round-off errors in floating-point can result in nonzero syndromes
  * Must select $\delta$ and only syndrome $> \delta$ will indicate an error
  * Value of $\delta$: probability of fault detection vs. false alarms
  * To simplify selection of $\delta$ - partition into submatrices with separate checksums