# Communications

**Jacky Baltes**

**University of Auckland**

**Email: j.baltes@auckland.ac.nz**

**Web: www.tcs.auckland.ac.nz/˜jacky**

**March 7, 2003**

## Error Detection Routines

- Parity Check: Add one bit to make the number of bits even/odd in a word.
- Assuming that a bit has a probability of $P_B$ of being wrong, what is the probability of: (a) receiving a byte with an error (8 bits), (b) receiving an error byte and not detecting an error.

$$P_a = (1 - P_B)^{N_B}$$

$$P_b = \sum_{j=2,4,\ldots}^{N_B} \left( \begin{array}{c} N_B \\ j \end{array} \right) P_B^j (1 - P_B)^{N_B - j}$$

- Errors are often not independent. Bursts of errors, especially at high transmission rates
- Vertical and longitudinal error checks
- Parity computed as exclusive or
- Can this method detect all errors?

## Cyclic Redundancy Checks

- Message $M$ ($n$ bits), and $k$ frame bits
- Choose $N$ ($k$ Bits) such that $M|N$ is exactly divisible by a predetermined number $P$ ($n+1$ Bits).
- How do we choose $N$?
- $$T = 2^n * M + N$$
  $$T/P = \frac{2^n * M + N}{P} = \frac{2^n * M}{P} + \frac{N}{P}$$
  $$= Q + R/P + \frac{N}{P}$$
  $$N = -R$$
- In Modulo arithmetic R = -R
- Sender: Compute $(2^n M)/P$ and add remainder as checksum
- Receiver: $T/P$, check if remainder is 0
- P is one bit longer than the checksum. Most and least significant bit should be one.
- CRC12: 12,11,3,2,1,0

- CRC16, 16,15,2,0
- Implement as shift/xor logic