Announcements

- Project phase 1 is posted
  - Due April 1, 20-min presentation in class and a written report.
  - When simulating, use a 127-bit PR sequence
Project

Channel: S21
Normalized SBR @ 6.25Gb/s 2-PAM & 4-PAM Nyquists

2-PAM SBR

4-PAM SBR

2-PAM 6.25Gb/s with Tx, Rx Eq

Tx Eq only

Tx + Rx Eq
4-PAM 6.25Gb/s with Tx, Rx Eq (prbs sequence)

Error Control

➢ Textbooks
  ➢ Lin, Costello, “Error Control Coding”
  ➢ Wicker, “Error Control Systems”
  ➢ Houghton, “Error Control for Engineers”

➢ Basics in e.g.
  ➢ Lee, Messerschmitt, “Digital Communications”
  ➢ Proakis, “Digital Communications"
## Error Control

### Error Detection
- The receiver has the capability to **detect** any code block that contains fewer than a predetermined number of symbols in error.

### Forward Error Correction
- A system of error control for data transmission wherein the receiver has the capability to **detect and correct** any code block that contains fewer than a predetermined number of symbols in error.

### Transmitter has to add redundancy to the transmitted data
- This defines the code rate
  - $k$ - user bits
  - $n$ - channel bits
  - $r$ - redundancy bits, $r = n - k$
  - $R = k/n$ – code rate

### Example: parity check in RS-232
- 8 bits of data are replaced with 9-bit codewords
  - 256 symbols are chosen out of a set of 512 that satisfy the property of having even parity
  - $R = 8/9$
Hamming Distance

- Minimum number of bits that has to be flipped in one codeword to get a valid codeword
- $d_{\text{min}} = 2$ in parity
- If we receive a message we can detect that it is invalid, but we have no way to figure out which one was sent
- There are 9 nearest neighbors with $d = 1$

Error Detection vs. Error Correction

- To detect $t$ errors:
  - $d_{\text{min}} > t$
  - $t = 1$ for parity code ($d_{\text{min}} = 2$)
- To correct $t$ errors:
  - $d_{\text{min}} > 2t$
  - Parity code cannot correct errors
  - Need $d_{\text{min}} = 3$: the correct symbol is one bit away, others are at least 2 bits away
Hamming Bound

- There is a total of $2^n$ symbols, $2^k$ are valid
- How to maximize $d_{\text{min}}$?
- Number of neighbors with distance $d$:
  \[
  \frac{n!}{d!(n-d)!}
  \]
- If the code corrects $t$ errors
  \[
  \sum_{d=0}^{t} \frac{n!}{d!(n-d)!} \leq 2^{n-k}
  \]

Hamming Codes

- There are some codes that exactly satisfy Hamming bound, or perfect codes $(n, k, t)$:
  - $(3, 1, 1), (7, 4, 1), (15, 11, 1)$
  - Hamming codes: $(2^m - 1, 2^m - m - 1, 1)$
  - Golay code: $(23, 12, 3)$ – corrects up to 3 errors
Perfect codes

- No extra symbols in $2^n$ symbol space
- Every symbol is remapped to a valid symbol
- No ability to detect more than $t$ errors
- E.g. Golay code $\langle 23, 12, 3 \rangle$ can be extended to $\langle 24, 12, 3 \rangle$ by adding a parity bit

Gilbert Bound

- The smallest symbol space $(2n)$ that guarantees the existence of a $t$-error correcting code with $k$ user bits

$$\sum_{d=0}^{2t} \frac{n!}{d!(n-d)!} \leq 2^{n-k}$$

- Most codes are between the Hamming and Gilbert bounds
  - If a message has more if $t$ detected errors, it will not be recovered (but will be flagged)
Some Coding Terms

- Block codes are memoryless
  - Combinatorial mapping
- Systematic codes include the user data and add some redundancy
- Convolutional codes have finite memory
  - We introduced them before Viterbi decoders

Syndrome

- \( n - k \) decoded redundant bits
- If no errors syndrome is 0
- If there are errors, syndrome points to bit position(s)
- To find a position of one error in 15 bits, need 4 extra bits \(- (15, 11, 1) \) code
Coding Gain

- Example: DVB-S2 (satellite TV broadcast)
  - DVB-S1 uses QPSK
  - DVB-S2 – more programs, HDTV, in the same band, same dish, same satellites
  - 8-PSK loses 2.5dB in SNR at required BER, to increase spectral efficiency by 50%
  - Forward error correction has to pick up the BER loss better than 2.5dB loss in SNR
## Error types

- **Random errors**
  - Additive white Gaussian noise (AWGN)

- **Error bursts**
  - Timing errors (e.g. PLL cycle slip, loss of lock, ...)
  - Supply disturbances
  - Thunders
  - Thermal asperities in disk drives
  - Scratches on a CD, DVD

- **Bursts are randomized by using interleaving**

## Error Detection Schemes

- **Parity**

- **Two-dimensional (horizontal/vertical) parity**
  - Vertical parity (checksum) is added at the end of a block
    
    |   |   |   |   |   |   |   |   |
    |---|---|---|---|---|---|---|---|
    | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
    | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
    | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
    | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
    
    Detected parity: 10110111
    Encoded parity: 10010110

- **Cyclic redundancy check (CRC)**
Coding Overhead

- Byte-level parity adds 11% of overhead for detecting one error
- This expands the required bandwidth (depending on the modulation scheme)
- Increases in-band noise

![Graph showing BER vs SNR for Coded and Uncoded signals]

Error Detection Schemes

- Cyclic redundancy check
- Symbol (message) bits are $d_i$, $i = 1, m - 1$.
- Add $n$ bits such that the complete message is divisible by a generator polynomial (GP) in the GF(2n)
  \[ CRC = \sum_{i=n}^{m-1} d_i \alpha^i \]
- The complete message
  \[ \sum_{i=0}^{m-1} d_i \alpha^i = 0 \]
CDC Detection Performance

- Usually a 2-byte (16-bit) CRC is used with 1k-2k blocks
- 0.8% - 1.6% overhead
- Detects all error bursts shorter than the length of GP
- Errors can be corrected as well

Error Correction

- Correction by parity
  - More about advanced parity checks later
- Correction with CRC
- Reed-Muller codes
  - Low rate
- Reed-Solomon codes
  - Most common in practice
Reed-Solomon Codes

- Invented ~1960
- A special case of BCH (Bose-Chaudhury-Hocquenghem) codes
- Starts with bits, $d_i$
  \[
  \sum_{i=0}^{m-1} d_i \alpha^i = 0
  \]
- And replaces with symbols, $p_i$
  \[
  \sum_{i=0}^{m-1} p_i \alpha^i = 0
  \]