CRC (Cyclic Redundancy Check) - Study Notes

Overview

CRC is a **powerful error detection method** widely used in real-life applications. It's the most commonly used error detection technique in practical environments.

Why CRC is Powerful

CRC can detect:

- **All odd errors** (any odd number of bit errors)
- **Single bit errors**
- **Double bit errors**
- **Burst errors** of length equal to polynomial degree

Key Concepts

Basic Formula

- **Total bits sent = m + r**
  - m = number of message bits
  - r = number of redundant bits (CRC bits)

Steps to Calculate CRC

Step 1: Determine Number of Redundant Bits (r)

Case 1: Polynomial Form Given

- If divisor is given as polynomial (e.g., $x^4 + x^3 + 1$)
- Number of redundant bits = **highest degree of polynomial**
- Example: $x^4 + x^3 + 1$ → degree = 4 → append 4 zeros

Case 2: Binary Form Given

- If divisor is given directly in binary (e.g., 11001)
- Number of redundant bits = **number of bits in divisor - 1**
- Example: 11001 has 5 bits → append 5-1 = 4 zeros

Step 2: Convert Polynomial to Binary (if needed)

Method: Extract coefficients

- Example: $x^4 + x^3 + 1$

- Coefficients: $x^4(1)$, $x^3(1)$, $x^2(0)$, $x^1(0)$, $x^0(1)$
- Binary form: **11001**

Step 3: Append Zeros to Message

- Append the required number of zeros to the original message
- Example: Message 1010011010 + 4 zeros = **10100110100000**

Step 4: Perform Binary Division Using XOR

XOR Rules:

- Same values → 0 ($1 \oplus 1=0$, $0 \oplus 0=0$)
- Different values → 1 ($1 \oplus 0=1$, $0 \oplus 1=1$)

Division Process:

1. Always start from leading 1
2. Use XOR operation (not regular division)
3. Continue until no more bits can be processed
4. **Quotient is ignored** - only remainder matters

Step 5: Replace Appended Zeros with Remainder

- Take the **last r bits** of the remainder (LSB side)
- Replace the appended zeros with these remainder bits
- This creates the **valid codeword** to be transmitted

Worked Example

Given:

- Message: 1010011010
- Divisor polynomial: $x^4 + x^3 + 1$

Solution:

1. Convert divisor: $x^4 + x^3 + 1$ → 11001
2. Append 4 zeros: 1010011010**0000**
3. Divide using XOR operations
4. Remainder: 000010 → Take last 4 bits: **0010**
5. Final codeword: 1010011010**0010**

Error Detection at Receiver

1. Receiver divides the entire received codeword by same divisor
2. **If remainder = 0** → No error detected
3. **If remainder ≠ 0** → Error detected

Efficiency Calculation

Formula: Efficiency = $(m/(m+r)) \times 100\%$

Where:

- m = message bits
- r = redundant bits

Example: 10 message bits + 4 redundant bits = $(10/14) \times 100\% = 71.43\%$

Important Tips for Exams

Common Mistakes to Avoid

- Don't confuse polynomial degree with number of binary digits
- Always start division from leading 1
- Take remainder from LSB side, not MSB
- Remember: quotient is not used in CRC

Quick Method for Division

- Always ensure first bit is 1 before applying divisor
- Carry bits from above when needed to maintain divisor length
- Use XOR operations consistently

Key Points to Remember

1. **Polynomial given** → append (highest degree) zeros
2. **Binary given** → append (number of bits - 1) zeros
3. **Division** → use XOR, start with leading 1
4. **Remainder** → take last r bits from LSB side
5. **Detection** → remainder = 0 means no error

Applications

- Most widely used in real-life networking applications
- Standard method in computer networks
- Used in data storage systems

- Common in communication protocols

Note: CRC numerical problems are very common in GATE and UGC-NET exams. Master the calculation steps as theory questions are rare (99% numerical, 1% theory).