Chapter 4
Digital Data Communication Techniques

Raj Jain
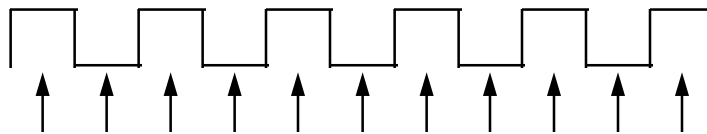
**Raj Jain is now at
Washington University in Saint Louis
Jain@cse.wustl.edu
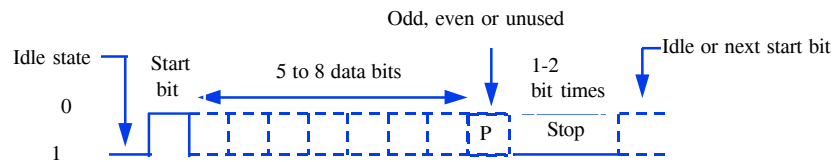http://www.cse.wustl.edu/~jain/**

Raj Jain

Overview

❑ Asynchronous vs Synchronous Transmissions

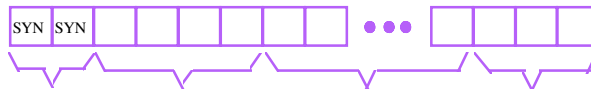❑ Error Detection: Parity, LRC, CRC

Raj Jain

Clock Synchronization

❑ Suppose, data rate = 10 kbps
One bit = 0.1 ms

❑ Clock rate is 1% faster,
Sampling every 0.099 ms

❑ After 50 bits: 50% away from center ⇒ Error

Asynchronous vs Synchronous

Odd, even or unused

Idle state   Start bit   5 to 8 data bits   1-2 bit times   Idle or next start bit

0

1

P   Stop

Character format

SYN SYN •••

One or more SYN character   Control characters   Data characters   Control characters

F   F

8-bit flag   Control fields   Data field   control field   8-bit flag
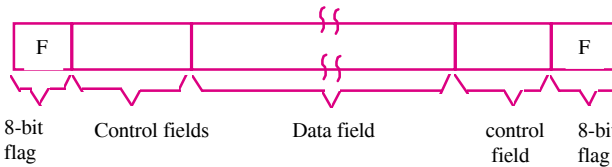
Asynchronous Transmission

❑ Used for short bit sequences

❑ Idle = No signal, negative voltage, 1

❑ One Start bit, 7 or 8 data bits

❑ One parity bit: Odd, Even, None

❑ Minimum Gap = Stop bits = 1, 1.5, or 2 bits

❑ Efficiency = data bits/total bits
8N1 = 1 Start bit + 8 Data bits + 1 Stop bit + 1 parity bit (even though the parity is not being used by this site)
$\Rightarrow 8/(1+8+1+1) = 73\%$

❑ Faster clock: $7\% \Rightarrow 56\%$ off on 8th bit $\Rightarrow$ Error

❑ Framing error $\Rightarrow$ False start/end of a frame

Synchronous Transmission

❑ Used for longer bit sequences

❑ Requires clock transmission
Use codes with clock information (Manchester)

❑ Begining of block indicated by a preamble bit pattern called "Syn"

❑ End of block indicated by postamble bit pattern

❑ Character-oriented transmission: Data in 8-bit units

❑ Bit-oriented transmission: Preamble = Flag

❑ Efficiency: Data bits/(Preamble+Data+Postamble)

❑ High-Level Data Link Control (HDLC) uses bit-oriented synchronous transmission. 8 bit of overhead for 1000 data bits
$\Rightarrow 48/1048 = 4.6\%$ overhead

Probability of Frame Error

- Probability of bit error = p
- Number of bits per frame = n
- Probability of no error in a bit = $1-p$
- Probability of i errors in a frame = $^nC_i \, p^i \, (1-p)^{n-i}$
  $^nC_i = n!/\{i! \, (n-i)!\}$
- Probability of no error in a frame = $(1-p)^n$
- Probability of error in a frame = $1-(1-p)^n$

| p | Error | p | Error | • • • | p | Error |
| 1-p | No error | 1-p | No error | | 1-p | No error |
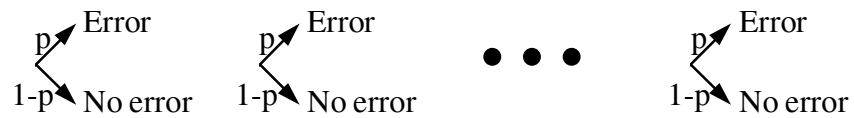
Error Detection Techniques

- Parity Checks
- Longitudinal redundancy checks
- Cyclic redundancy checks

Parity Checks

1	0	1	1	1	0	1	0	
1 2 3 4 5 6 7 8 9

❑ Odd Parity

N

| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
1 2 3 4 5 6 7 8 9   1 2 3 4 5 6 7 8 9

1-bit error

| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
1 2 3 4 5 6 7 8 9   1 2 3 4 5 6 7 8 9

3-bit error   2-bit error

❑ Even Parity

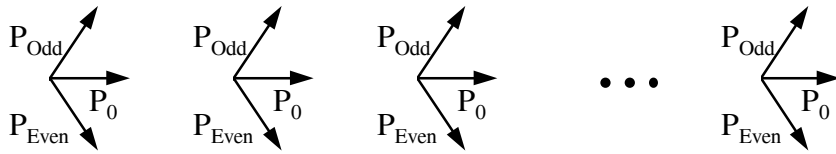| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
1 2 3 4 5 6 7 8 9

Probability of Error with Parity

❑ Let, n= Number of bits per character (= N_B in the book)
  n includes the parity bit
  N=number of characters per frame (=N_C in the book)

❑ Probability of no errors/Char $P_0 = (1-P_B)^n$
  P_B = bit error probability = p

❑ Probability of i bit errors/Char = $^nC_i\, P_B^i (1-P_B)^{n-i}$

❑ Probability of odd # of bits errors/Char
  $P_{Odd} = n\, PB(1-P_B)^{n-1} + {}^nC_3\, P_B^3(1-P_B)^{n-3} + {}^nC_5\, P_B^5(1-P_B)^{n-5} +$

❑ Probability of undetected (even # of bits) errors/Char
  $P_{Even} = {}^nC_2\, P_B^2(1-P_B)^{n-2} + {}^nC_4\, P_B^4(1-P_B)^{n-4} + {}^nC_6\, P_B^6(1-P_B)^{n-6} +$
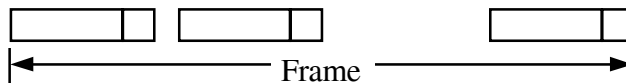
⟵———————— Frame ————————⟶

Probabilities (Cont)

- Probability of no errors/frame = $P_1 = (1-P_B)^{nN}$
- P(Undetected Error) = P_{UE}
  = P(k even errors and N-k no errors)
  = $\Sigma^N_{k=1} {}^N C_k P_{Even}^{\ k} P_0^{\ (N-k)}$
- Probability of detected errors P_{DE}
  = $1-P(\text{no errors}) - P(\text{undetected errors})$
  = $1- P_1 - P_{UE}$

P_{Odd} P_0 P_{Even} P_{Odd} P_0 P_{Even} P_{Odd} P_0 P_{Even} $\cdots$ P_{Odd} P_0 P_{Even}

Longitudinal Redundancy Checks (LRC)

	bit 1	bit 2		bit n	parity bit
Character 1	b_{11}	b_{21}		bn_1	R_1
Character 2	b_{12}	b_{22}		bn_2	R_2
Character m	b_{1m}	b_{2m}		b_{nm}	R_m
Parity check character	C_1	C_2		C_n	C_{n+1}

Vertical parity

VRC

```
1 0 1 1 0 1 1 1
1 1 0 1 0 1 1 1
0 0 1 1 1 0 1 0
1 1 1 1 0 0 0 0
1 0 0 0 1 0 1 1
0 1 0 1 1 1 1 1
0 1 1 1 1 1 1 0  ← LRC
```

← Longitudinal parity
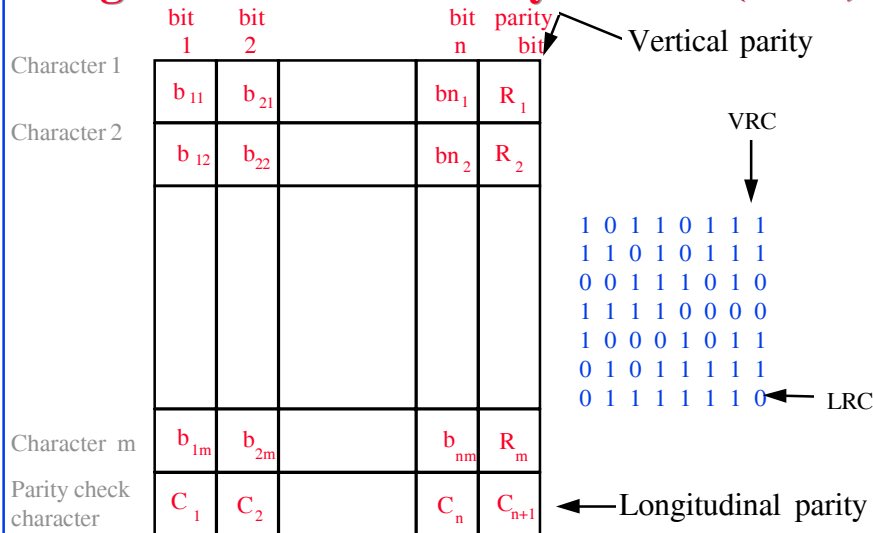
Check Digit Method

❑ Make number divisible by 9

Example: 823 is to be sent

1. Left-shift: 8230

2. Divide by 9, find remainder: 4

3. Subtract remainder from 9: 9-4=5

4. Add the result of step 5 to step 1: 8235

5. Check that the result is divisible by 9.

Detects all single-digit errors: 7235, 8335, 8255, 8237

Detects several multiple-digit errors: 8765, 7346

Does not detect some errors: 7335, 8775, ...

Modulo 2 Arithmetic

```
 1111     11001        1100
+1010   ×    11    11 | 10101
                      / 11
--------  --------   ----
 0101     11001       x11
          11001        11
                     ----
         ---------    x00
         101011       00
                     ----           010   2
                      x01           011   3
                       00           ----   --
                      ----          001   1  Mod 2
                       x1           101   5  Binary
```

Cyclic Redundancy Check (CRC)

❑ **Binary Check Digit Method**

❑ Make number divisible by P=110101 (n+1=6 bits)

Example: M=1010001101 is to be sent

1. Left-shift M by n bits 2^nM= 101000110100000

2. Divide 2^nM by P, find remainder: R=01110

~~3. Subtract remainder from P~~

4. Add the result of step 2 to step 1 : T=101000110101110

5. Check that the result T is divisible by P.

Detects all single-bit errors

Detects several multiple-bit errors

Does not detect some errors

Modulo 2 Division

```
      Q=1101010110
 P=110101)101000110100000=2ⁿM
        110101
        111011                010110
        110101                000000
         011101                101100
         000000               110101
          111010                110010
          110101                110101
           011111                001110
           000000                000000
            111110                01110 = R
            110101
```

Checking At The Receiver

```
          1101010110
 110101)101000110101110
        110101
         111011             010111
         110101             000000
          011101             101111
          000000             110101
           111010             110101
           110101             110101
            011111             00000
            000000
             111110
             110101
```

Polynomial Representation

❑ Number the bits 0, 1, ..., from right

$b_n b_{n-1} b_{n-2}....b_3 b_2 b_1 b_0$

$b_n x^n + b_{n-1} x^{n-1} + b_{n-2} x^{n-2} + ... + b_3 x^3 + b_2 x^2 + b_1 x + b_0$

❑ Example:

543210
↓↓↓↓↓

$110101 = x^5 + x^4 + x^2 + 1$

$1101\ 1001\ 0011 = x^{11} + x^{10} + x^8 + x^7 + x^4 + x + 1$

11 10 9 8   1   0

Cyclic Redundancy Check (CRC)

Polynomial Division Method

Make $T(x)$ divisible by $P(x) = x^5+x^4+x^2+1$ (Note: $n=5$)

Example: M=1010001101 is to be sent

$M(x) = x^9+x^7+x^3+x^2+1$

1. Multiply M(x) by x^n, $x^nM(x) = x^{14}+x^{12}+x^8+x^7+x^5+ \ldots$

2. Divide $x^nM(x)$ by $P(x)$, find remainder:

$R(x)=01110=x^3+x^2+x$

4. Add the remainder $R(x)$ to $x^nM(x)$:

$T(x)= x^{14}+x^{12}+x^8+x^7+x^5+x^3+x^2+x$

5. Check that the result $T(x)$ is divisible by $P(x)$.

Transmit the bit pattern corresponding to $T(x)$:

101000110101110

Popular CRC Polynomials

❏ CRC-12: $x^{12}+x^{11}+x^3+x^2+x+1$

❏ CRC-16: $x^{16}+x^{15}+x^2+1$

❏ CRC-CCITT: $x^{16}+x^{12}+x^5+1$

❏ CRC-32: Ethernet, FDDI, ...

$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}$
$\qquad +x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

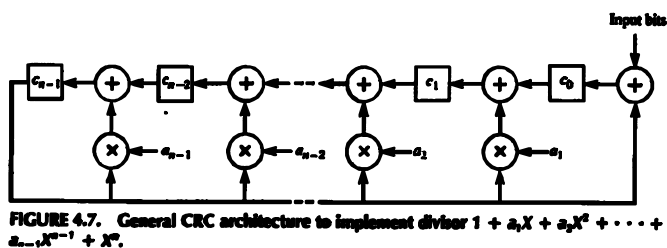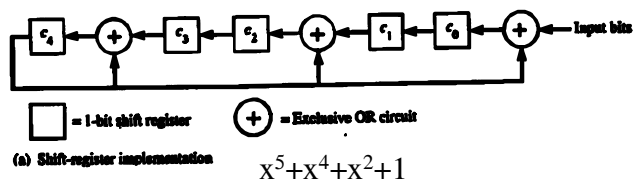Even number of terms in the polynomial

$\Rightarrow$ Polynomial is divisible by 1+x

$\Rightarrow$ Will detect all odd number of bit errors

Errors Detected by CRC

❑ All single bit errors

❑ Any burst error of length n bits or less, n=degree of the polynomial

❑ Most larger burst errors
P(undetected burst errors|error has ocurred) = 2^{-n}

❑ Any odd number of errors if P(x) has 1+x as a factor, i.e., has even number of terms

❑ Any double bit errors as long as P(x) has a factor with 3 terms, e.g., $(1+x^4+x^9)(....)$

Shift-Register Implementation

$x^5+x^4+x^2+1$

FIGURE 4.7.   General CRC architecture to implement divisor $1 + a_1X + a_2X^2 + \cdots + a_{n-1}X^{n-1} + X^n$.

Error Correcting Codes (ECC)

❏ Example:
VRC+LRC will correct all single bit errors

❏ Forward error correction (FEC)
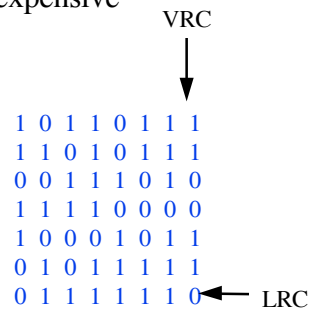Used if retransmission expensive

VRC

```
1 0 1 1 0 1 1 1
1 1 0 1 0 1 1 1
0 0 1 1 1 0 1 0
1 1 1 1 0 0 0 0
1 0 0 0 1 0 1 1
0 1 0 1 1 1 1 1
0 1 1 1 1 1 1 0  ← LRC
```

Fig 4.5b

Summary

❏ Asynchronous and Synchronous transmission
❏ Parity, LRC, VRC, CRC
❏ CRC Polynomials

Homework

- Exercise: 4.1
  In b assume flags are included in 48 control bits.
  In d, assume syn is included in 9 control characters.
  Assume 7 data bits + 1 parity.

- Exercise: 4.13ab
  P(Undetected error) = 1- {P(No error) + P(Detected Errors)}

- Exercises: 4.14, 4.16

- Due next class