

# End-to-End Data



# Presentation Layer

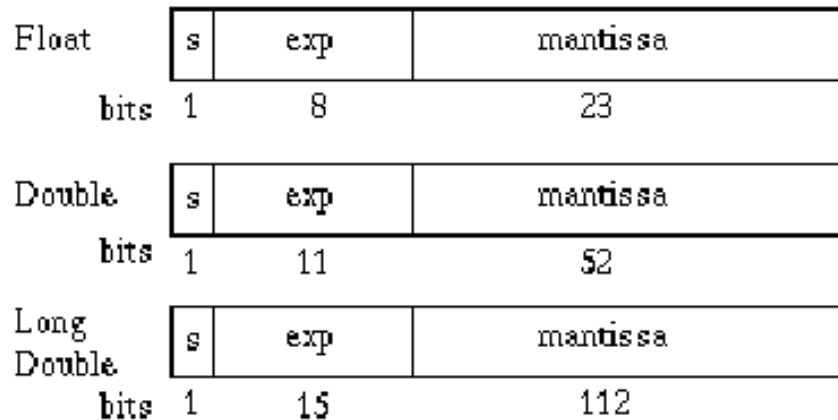
- Want to convey data to someone remotely
- Applications display data
- Need standard way to represent data so different applications can display it
- Data Encoding
  - ASN.1 Encoding
  - eXtensible Markup Language (XML)
- Multimedia Encoding
  - Image Compression
  - Audio/Video Codecs



# Simple Byte Ordering

- Integers
  - Big Endian
    - High order byte first
  - Little Endian
    - Low order byte first
  - Host-byte order vs Network-byte order

- Floating Point
  - IEEE 754



# More Complex Data Types

- Structures? Non-numeric?
- Abstract Syntax Notation One (ASN.1)
  - ISO Standard
  - Basic Encoding Rules (BER)
- Format:
  - Tag = 8 bits
  - Length = length of Value (either 1 byte or 32 bytes, depending on length of value)
  - Value = opaque block of data, interpreted per “Tag”



# Markup Languages

- Most Common: HTML
  - Used by web browsers
  - Tags define text properties
    - Intermingled with data
- Broader: XML



# XML Example

```
<recipe name="bread" prep_time="5 mins" cook_time="3 hours">
  <title>Basic bread</title>
  <ingredient amount="3" unit="cups">Flour</ingredient>
  <ingredient amount="0.25" unit="ounce">Yeast</ingredient>
  <ingredient amount="1.5" unit="cups" state="warm">Water</ingredient>
  <ingredient amount="1" unit="teaspoon">Salt</ingredient>
  <instructions>
    <step>Mix all ingredients together.</step>
    <step>Knead thoroughly.</step>
    <step>Cover with a cloth, and leave for one hour in warm room.</step>
    <step>Knead again.</step>
    <step>Place in a bread baking tin.</step>
    <step>Cover with a cloth, and leave for one hour in warm room.</step>
    <step>Bake in the oven at 350(degrees)F for 30 minutes.</step>
  </instructions>
</recipe>
```



# XML Schema and Namespaces

- Schema
  - Defines valid tags, and data type
  - Allows syntax checking on XML
  - Schema is an XML file using XMLSchema Schema
- Namespaces
  - Names not necessarily unique between schemas
  - Use two schemas together
  - Need a way to uniquely refer to each



# Data Compression

- Representation of data may involve significant redundancy
- Compress data use less storage/network resources
- Lossless
  - Able to exactly reconstruct original data given the compressed form
- Lossy
  - Able to closely reconstruct original data given the compressed form

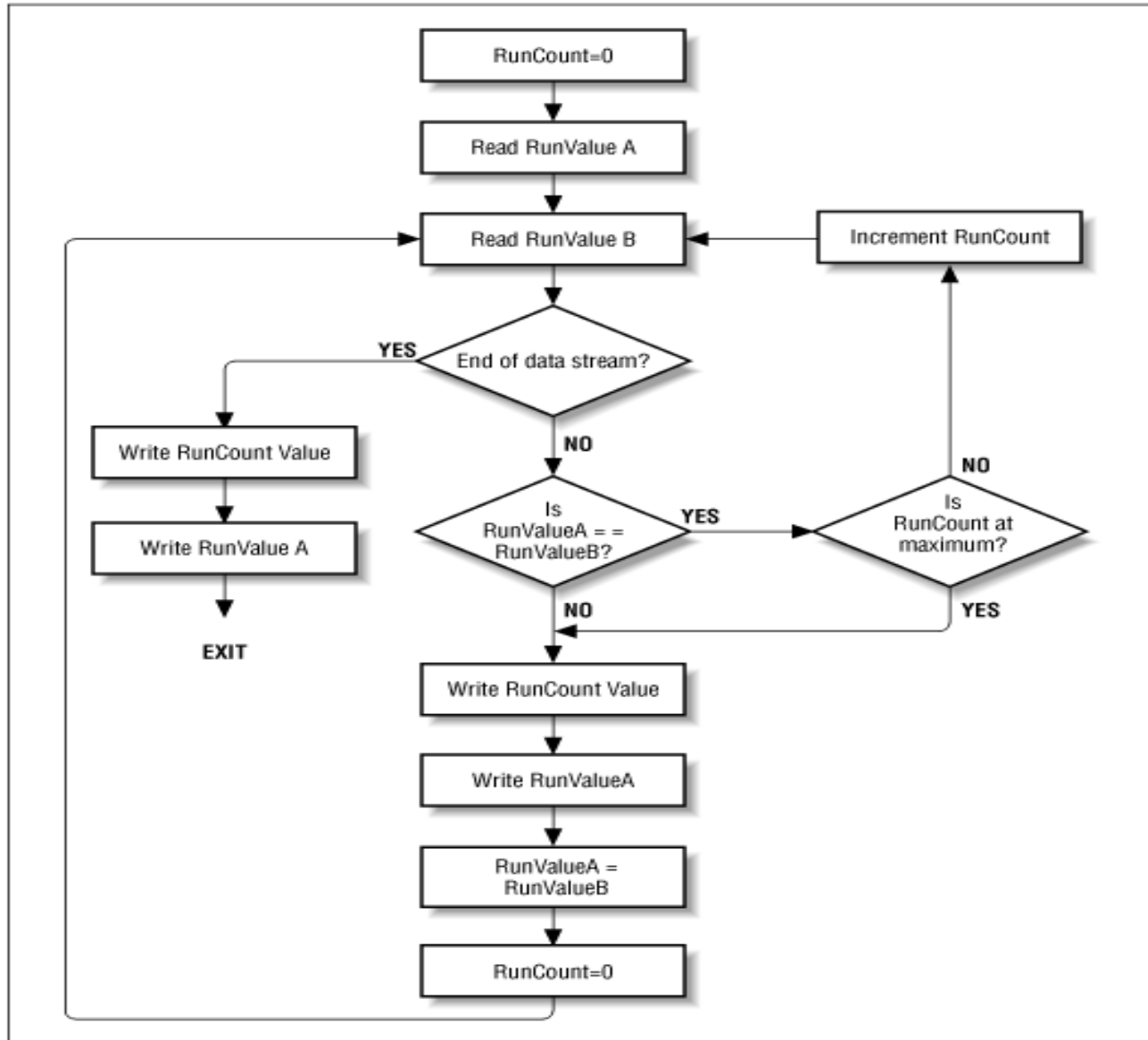


# Run-Length Encoding

- Early bitmap image compression
- AABBCDDDD = 3A2B1C4D
- Works well on scanned documents
  - Black and white – only two colors
  - Lots of white space



# Run-Length Encoding



# Differential Pulse Code Modulation

- DPCM
- AABBCDDDD
- A = reference symbol
- Encoded: A0001123333
  - Values 0-3 can be represented as 2 bits rather than 8 bits
- Works better than RLE for digital images
  - Slow changes in color values



# Dictionary-Based Methods

- Lempel-Ziv (LZ) compression
- Dictionary:
  - List of indexed words
- In source document, replace instances of dictionary words with index into dictionary
- Dictionary source?
  - Data-specific
  - GIF compression: indexed colors



# JPEG Image Compression

- Lossy
- Breaks image up into squares, quantizes and compresses each square
  - Least significant data removed from each block
  - RLE encoding of quantized data in each block



# Interlaced GIF

- GIF 89a
- Low resolution version first, then higher resolution
- Slow data rate: see blurry version first, then clear version as the data loads

