Prof. Philip Koopman

Carnegie Mellon University

Electrical & Computer ENGINEERING

Data Integrity

~"I have a bad feeling about this."~

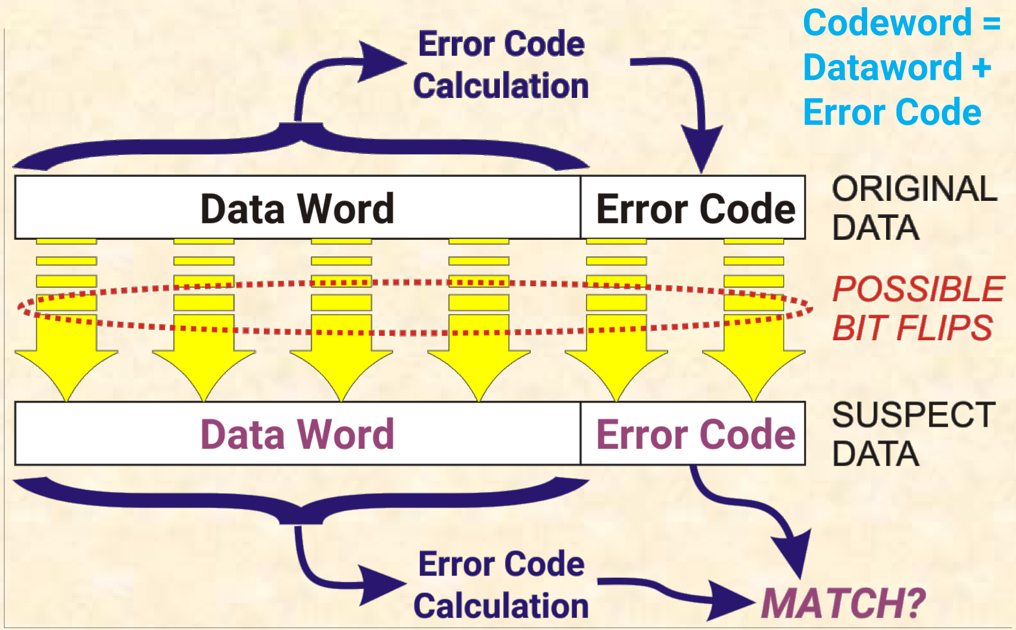
— Star Wars, Episode k {k=1..9}

These tutorials are a simplified introduction, and are not sufficient on their own to achieve system safety. You are responsible for the safety of your system.

Data, Message & Memory Integrity

Carnegie Mellon University

- **Anti-Patterns for Data Integrity:**
  - No checks on memory data
    - Program image and configuration
    - RAM and other data integrity
  - No end-to-end message checks
  - Using checksum instead of CRC



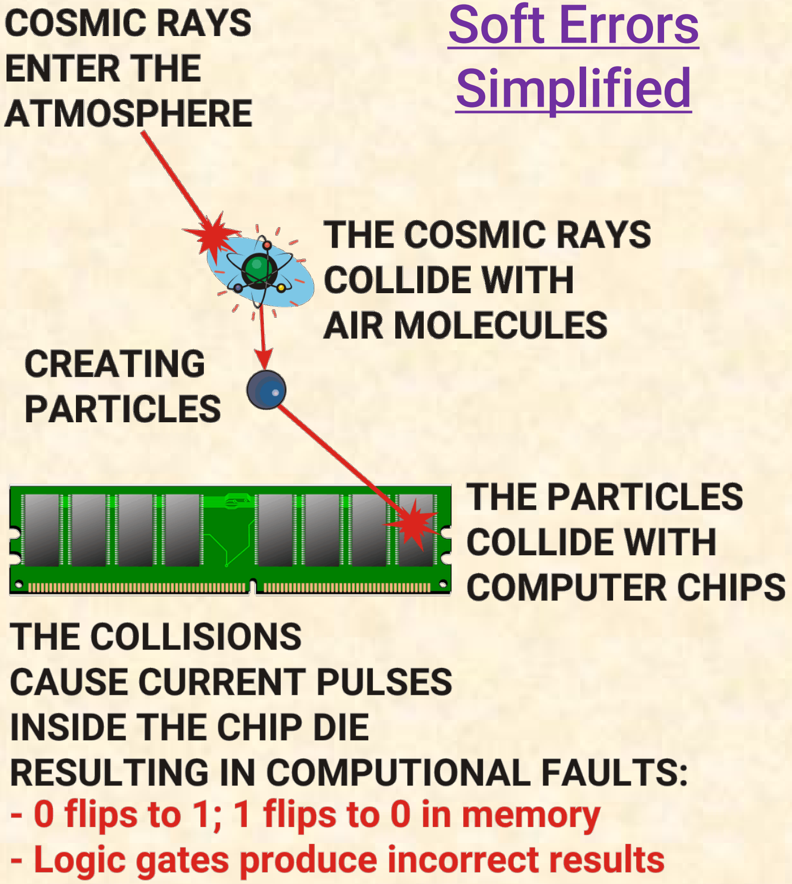
- **Memory & data integrity**
  - Detecting data corruption:
    - Mirroring, Parity & SECMED codes, Checksum, CRC
    - If data word consistent with error code, then no *detectable* error
    - Random hash as a starting point: random k-bit error code by chance misses $1/2^k$ errors
  - Malicious faults require cryptographically strong integrity check
    - All error codes discussed here are easy to attack

Sources of Data Faults

Carnegie Mellon University

- Hardware faults
  - Network message bit flips
  - Bad EEPROM/Flash writes
  - "Bit rot" (storage degrades over time)
- Single event upsets: Soft Errors
  - Affect both memory & CPU logic
  - Error detecting codes usually don't help with CPU logic faults!
- Software corruption
  - Bad pointers, buffer overflow, etc.

Soft Errors Simplified



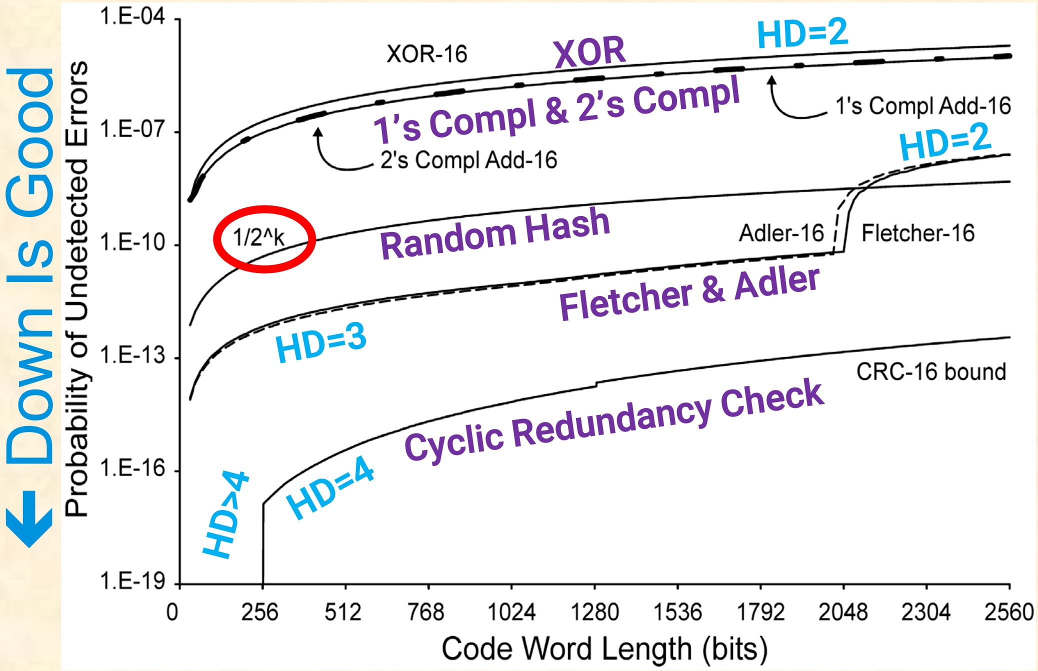
Overview of Data Integrity Mechanisms

- Key term: Hamming Distance (HD)
  - Smallest # of bit flips possibly undetected
  - Flips across data value and error code
  - Higher HD is better (more errors detected)
- Parity: detects single bit errors (HD=2)
  - Store one bit that holds XOR of all bits
- Mirroring (HD=2, but cheap computation)
  - Store data twice: plain and inverted bits
    - E.g.: 0x55 ➔ {0x55, 0xAA} two-byte pair
- SEC: (Hamming Code) correct single bit errors
- SECDED:
  Single Error Correction, Double Error Detection
  - Use a Hamming Code + parity bit to give HD=4
  - Size approximately 1 + $\log_2$ (number of data bits)

HD	Flips Detected	Flips Undetected	Examples
1	None	1+	No Error Code
2	1	2+	Parity, Checksum, Mirroring, Any CRC
3	1-2	3+	Hamming (SEC), Some CRCs, Short Fletcher
4	1-3	4+	Some CRCs, SECDED
5+	HD-1	HD+	Good CRC

Checksum Techniques Compared

- "Add" up all the data bits
  - XOR all data words (HD=2)
    - Detects 1-bit errors
  - 2's complement addition (HD=2)
    - Detects 1-bit and most 2-bit errors
  - 1's complement addition (HD=2)
    - Wraps carry bit, so slightly better
- Complex checksums:
  - Fletcher checksum (HD=2, HD=3)
    - Keeps two running 1's comp. sums
    - HD=3 at short lengths, HD=2 at long lengths
  - Adler checksum (HD=2, HD=3)
    - Uses prime moduli counters
    - Fletcher is typically a better & faster choice

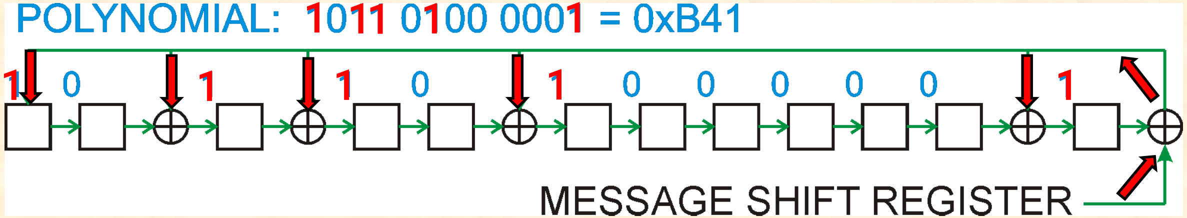


Maxino, T., & Koopman, P. "The Effectiveness of Checksums for Embedded Control Networks," IEEE Trans. on Dependable and Secure Computing, Jan-Mar 2009, pp. 59-72.

Error rate BER = 10^{-6}

Cyclic Redundancy Check (CRC)

- The mechanism:
  - Shift and XOR of selected feedback bits
  - Accumulated residue in shift register is the CRC "checksum" value



Example Feedback Polynomial:
0xB41 = $x^{12}+x^{10}+x^9+x^7+x+1$ ("+1" is implicit in hex value)
= $(x+1)(x^3+x^2+1)(x^8+x^4+x^3+x^2+1)$
Factor of (x+1) ➔ implicit parity (detects all odd errors)

- The math:
  - The data and the feedback bit pattern are both binary coefficient polynomials
  - Error code is <u>remainder</u> from polynomial division of data by feedback over GF(2)
- Feedback polynomial selection matters
  - Some popular polynomials are poor choices, including international standards(!)
  - Some rules of thumb are misguided (e.g., (x+1) divisibility for high HD)
  - Best polynomials are found via brute force search of exact evaluations

Finding "Good" Polynomials

Carnegie Mellon University

https://users.ece.cmu.edu/~koopman/crc/

- Example: HD=4 for 256 bit data word ➔ 0x247 (10 bit CRC)
- Example: HD=6 for 128 bit data word ➔ 0x9eb2 (16 bit CRC)

Max length at HD / **Polynomial**	**CRC Size (bits)**													
	3	4	5	6	7	8	9	10	11	12	13	14	15	16
HD=2	**0x5**	**0x9**	**0x12**	**0x33**	**0x65**	**0xe7**	**0x119**	**0x327**	**0x5db**	**0x987**	**0x1abf**	**0x27cf**	**0x4f23**	**0x8d95**
HD=3	4 **0x5**	11 **0x9**	26 **0x12**	57 **0x33**	120 **0x65**	247 **0xe7**	502 **0x119**	1013 **0x327**	2036 **0x5db**	4083 **0x987**	8178 **0x1abf**	16369 **0x27cf**	32752 **0x4f23**	65519 **0x8d95**
HD=4			10 **0x15**	25 **0x23**	56 **0x5b**	119 **0x83**	246 **0x17d**	501 **0x247**	1012 **0x583**	2035 **0x8f3**	4082 **0x12e6**	8177 **0x2322**	16368 **0x4306**	32751 **0xd175**
HD=5					4 **0x72**	9 **0xeb**	13 **0x185**	21 **0x2b9**	26 **0x5d7**	53 **0xbae**	52 **0x1e97**	113 **0x212d**	136 **0x6a8d**	241 **0xac9a**
HD=6						4 **0x9b**	8 **0x13c**	12 **0x28e**	22 **0x532**	27 **0xb41**	52 **0x1e97**	57 **0x372[illegible]**	114 **0x573a**	135 **0x9eb2**
HD=7								5 **0x29b**	12 **0x571**	11 **0xa4f**	12 **0x12a5**	13 **0x28a9**	16 **0x5bd5**	19 **0x968b**
HD=8									4 **0x4f5**	11 **0xa4f**	11 **0x10b7**	11 **0x2371**	12 **0x630b**	15 **0x8fdb**

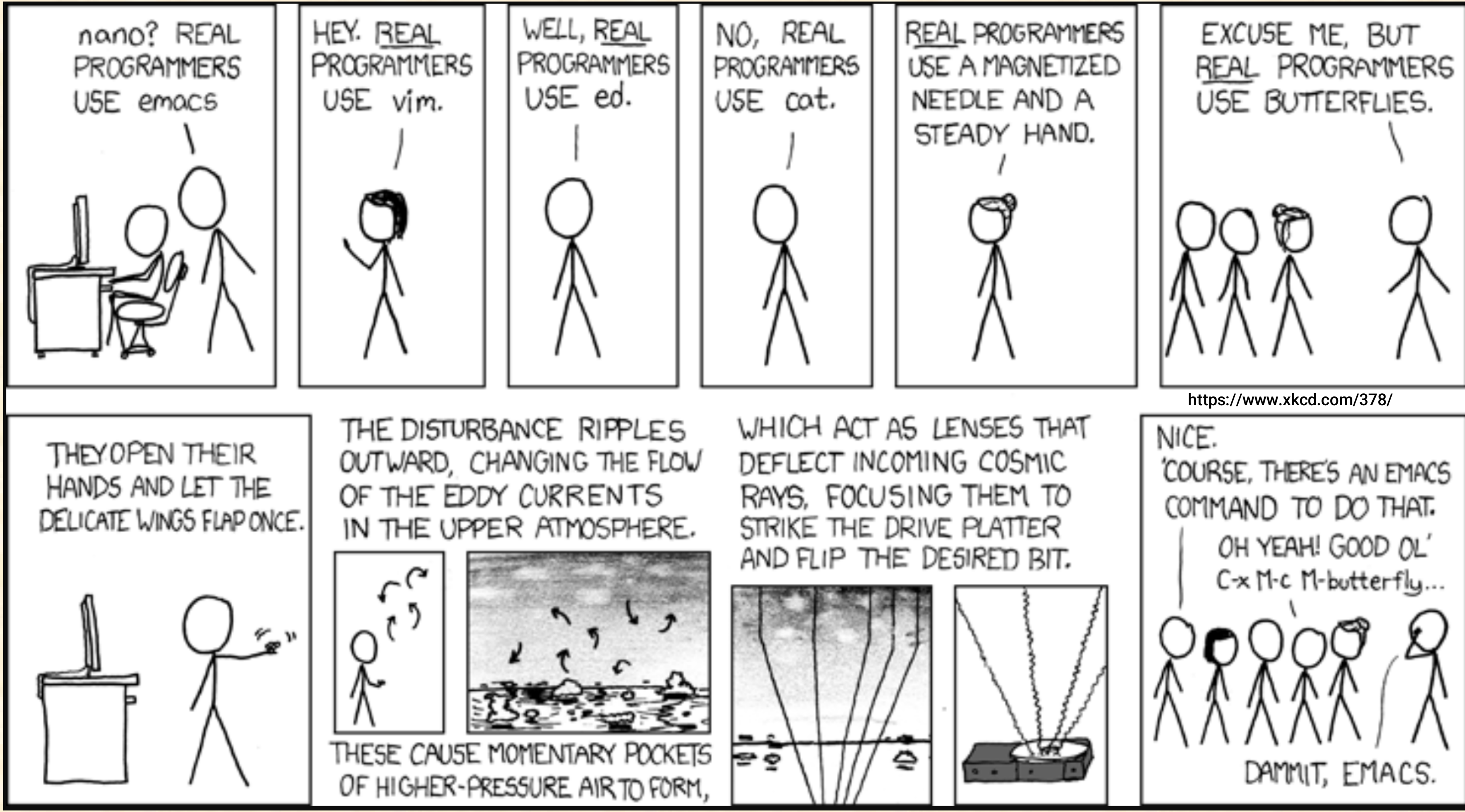
32
0xad0424f3
4294967263 **0xad0424f3**
2147483615 **0xc9d204f5**
65505 **0xd419cc15** (**)
32738 **0x9960034c** (**)
992 **0xf8c9140a** (**)
992 **0xf8c9140a**
223 **0x9d7f97d6**
100 **0xb49c1c96**

Best Practices For Data Integrity

Carnegie Mellon University

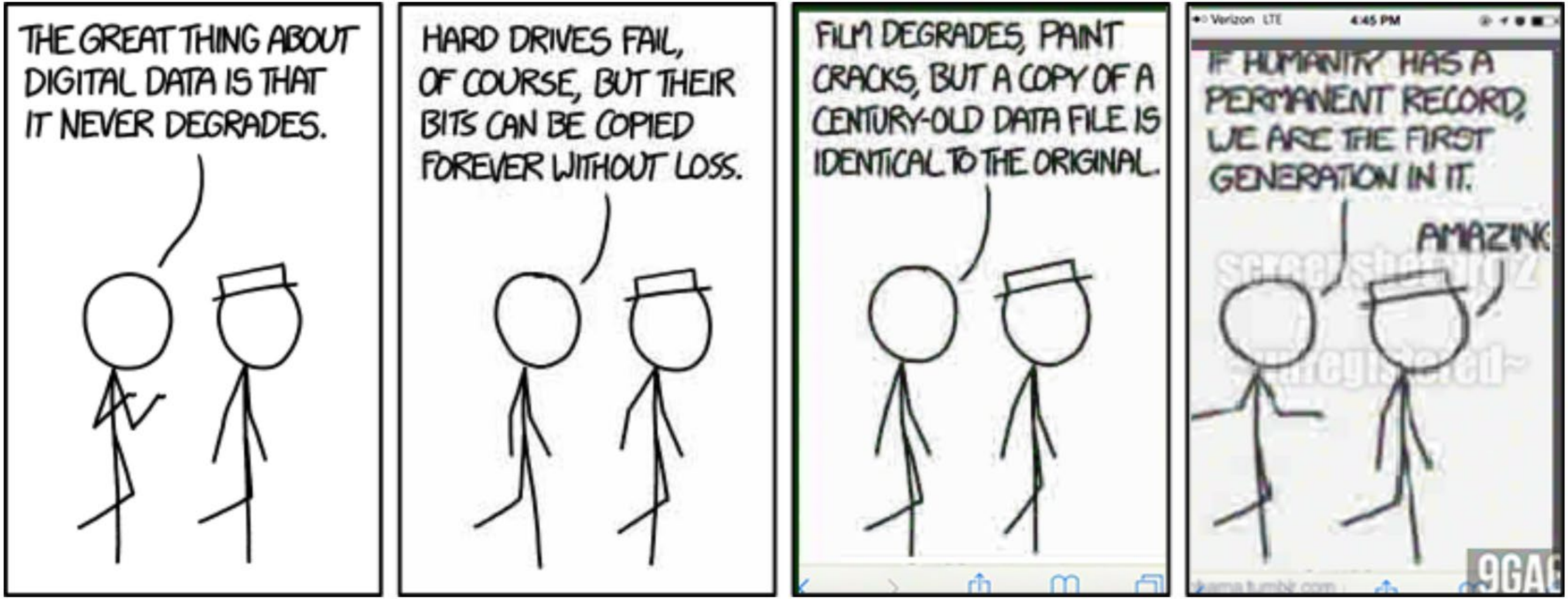
- Ensure sufficient data integrity
  - CRC on network packets
  - Periodic CRC on flash/EEPROM data
  - Appropriate memory integrity check on RAM

- Pitfalls:
  - Assuming mirroring is enough
    - What about data on stack?
    - What about data inside operating system?
  - Assuming memory data integrity is all you need
    - What about corrupted calculations?
  - Using a checksum when you should use a CRC
  - Many subtle pitfalls for the unwary. See FAA report: https://goo.gl/uKFmHr



https://www.xkcd.com/378/

Digital Data



Title text: â€œIf you can read this, congratulationsâ€”the archive youâ€™re using still knows about the mouseover textâ€!

https://www.explainxkcd.com/wiki/index.php/1683:_Digital_Data