Computer Science CSCI 355

Digital Logic and Computer Organization

Dr. Peter Walsh
Department of Computer Science
Vancouver Island University
peter.walsh@viu.ca

Codes

○ Numeric Coding

- e.g. 2's complement
- e.g. excess 127
- e.g. Grey Code

○ Character Coding

- e.g. ascii

○ Error Coding

- concerned with error detection and correction
- e.g. Hamming Codes

System Defects

○ Fault versus Error versus Failure

- fault is a flaw
- error is an observable difference between actual output and expected output

  Dijkstra: *program testing can only be used to show the presence of bugs and not their absence*
- failure is the inability of the system (or component) to perform its required function according to its specification

System Defects cont.

○ Software

- e.g. `B = 0; if (A > 2) [B = 4 ]`
- fault is that `>` should be `>=`
- test case to expose error

  `A = 2; error if (B != 4)`

○ Hardware

- e.g. an input to an AND gate is stuck at 0
- fault is that the gate input wire is somehow compromised
- test case to expose error

  set inputs to `1`; error if output `!= 1`

Data Communication Faults

○ Interference
- E.M. radiation

○ Distortion
- medium blocks some frequencies

○ Attenuation
- signal becomes weaker over long distances

Data Communication Errors

○ Types

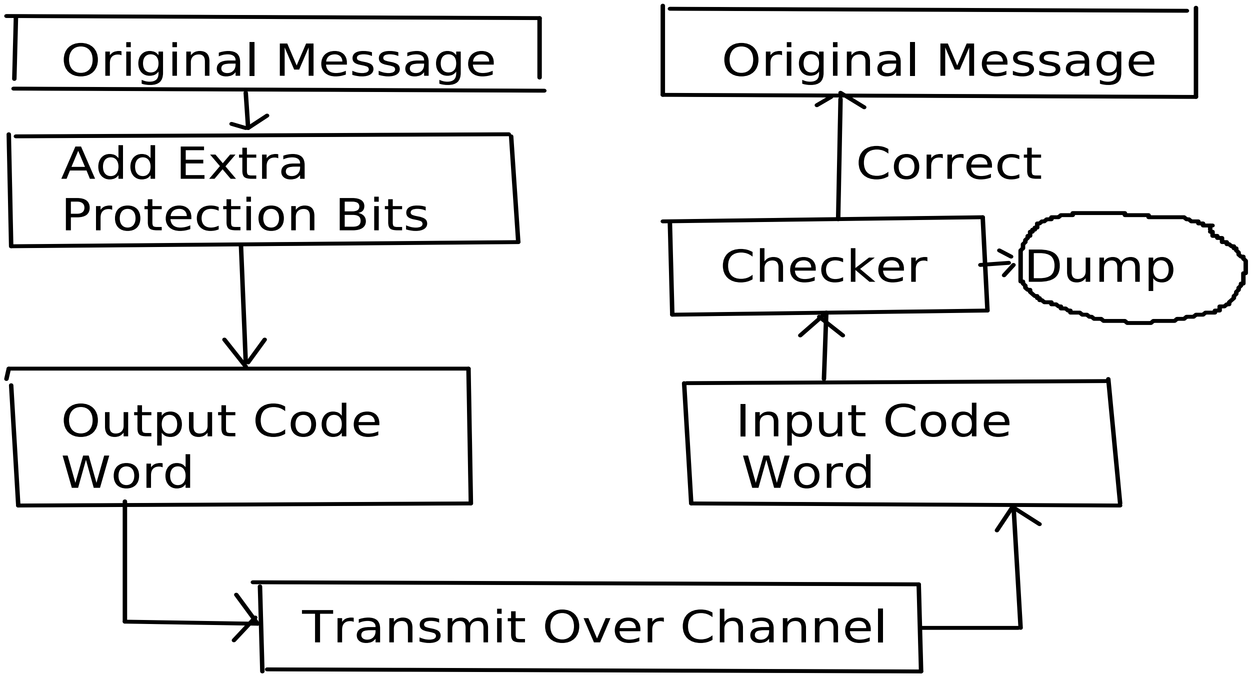
- single bit errors
- burst (multi bit) errors
- erasure (ambiguity)

○ CSCI 355 Focus

- transmission errors rather than faults

Error Detection/Correction

- Basic Idea



One Bit Error Detection

○ Even Parity

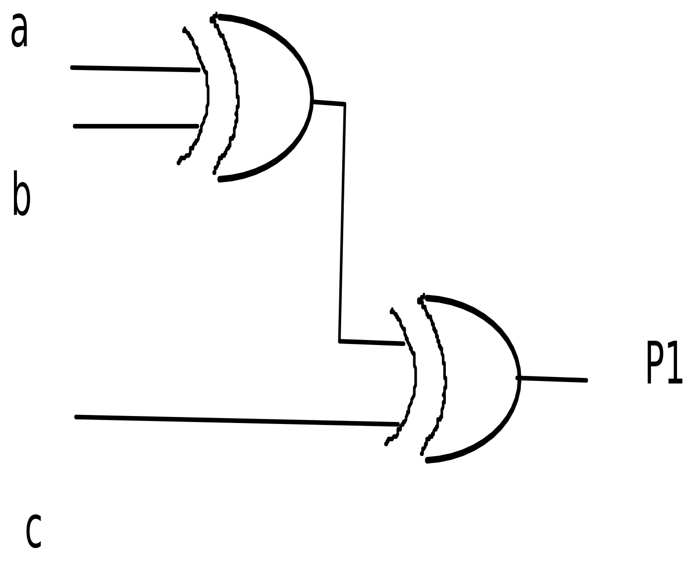
- e.g. 3 data bits, 1 protection (parity) bit

a	b	c	Even Parity P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

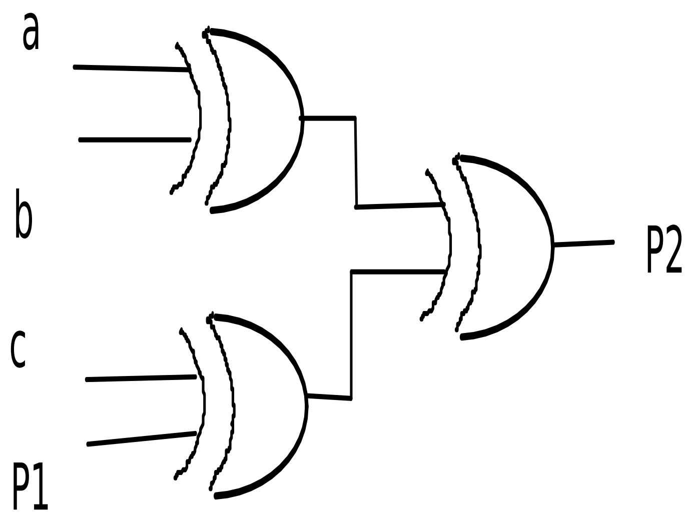
Code Words

a	b	c	P1	Code Word	P2
0	0	0	0	Y	0
0	0	0	1	N	1
0	0	1	0	N	1
0	0	1	1	Y	0
0	1	0	0	N	1
0	1	0	1	Y	0
0	1	1	0	Y	0
0	1	1	1	N	1
1	0	0	0	N	1
1	0	0	1	Y	0
1	0	1	0	Y	0
1	0	1	1	N	1
1	1	0	0	Y	0
1	1	0	1	N	1
1	1	1	0	N	1
1	1	1	1	Y	0

Even Parity Generator



Even Parity Checker



Hamming Codes (Error Correction)

○ Turing Award 1968 (Nobel Prize for Computing)

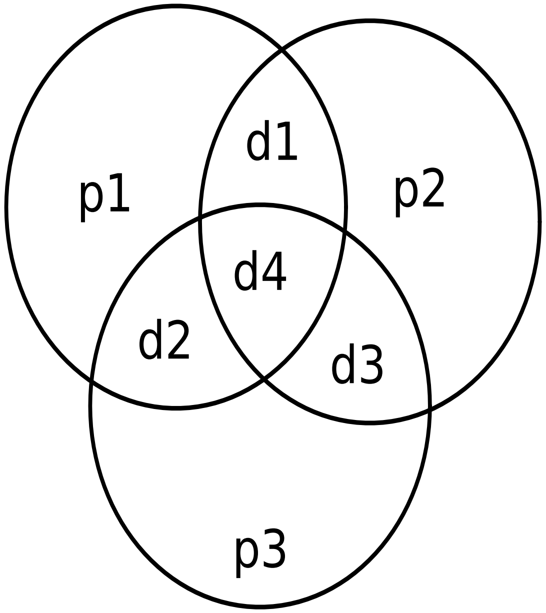
○ Basic Idea

- if 1 parity bit is good then more than 1 would be better

○ Method (Single Bit Error Correcting)

- n data bits and m check (parity bits) each of the m parity bits will protect a different combination of the n data bits

Who Covers What?



n=4 (d4 d3 d2 d1) m=3 (p1 p2 p3)

Parity Calculation

○ Generator

- $p1 = d1 \oplus d2 \oplus d4$
- $p2 = d1 \oplus d3 \oplus d4$
- $p3 = d2 \oplus d3 \oplus d4$

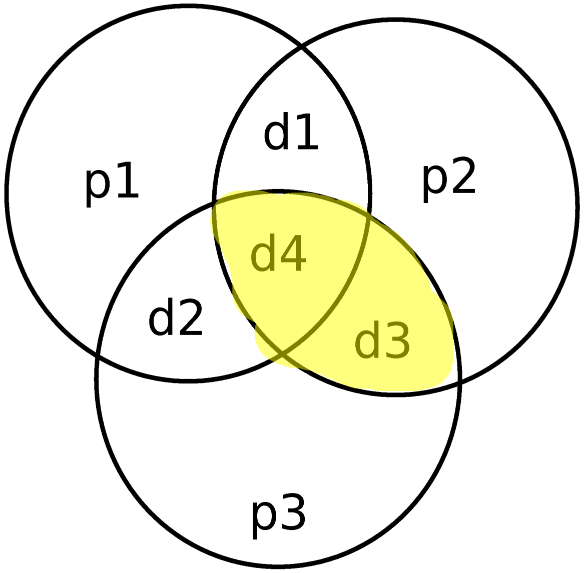
○ Checker

- $c1 = p1 \oplus d1 \oplus d2 \oplus d4$
- $c2 = p2 \oplus d1 \oplus d3 \oplus d4$
- $c3 = p3 \oplus d2 \oplus d3 \oplus d4$

○ Syndrome

- $c3\ c2\ c1$ collectively referred to as the syndrome

Single Bit Error (Which Bit?)



e.g. c3 =1, c2=1, c1=0
d3 is in error

Single Bit Error (Which Bit?) cont.

Syndrome c3	 c2	 c1	Inference
0	0	0	no error
0	0	1	p1 in error
0	1	0	p2 in error
0	1	1	d1 in error
1	0	0	p3 in error
1	0	1	d2 in error
1	1	0	d3 in error
1	1	1	d4 in error