COE 202: Digital Logic Design
Number Systems
Part 4

Dr. Ahmad Almulhem
Email: ahmadsm AT kfupm
Phone: 860-7554
Office: 22-324

Objectives

- Binary codes
  - Binary coded decimal (BCD)
  - Other Decimal Codes
  - Gray Code
  - ASCII Code
  - Error Detecting Code

Binary Codes

- A n-bit binary code is a binary string of 0s and 1s of size n.
- It can represent 2^n different elements.
  - 4 elements can coded using 2 bits
  - 8 elements can be coded using 3 bits
- Given the number of elements to be coded, there is a *minimum* number of bits, but no *maximum* !

Binary Coded Decimal (BCD)

- Human communicating with computers
  - Humans understand decimal
  - Computers understands binary
- Solution: Convert Decimal-Binary-Decimal
- Need to store decimal numbers as binary codes

Binary Coded Decimal (BCD)

Decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Convert 2496_{10} to BCD code:

2 4 9 6

↓ ↓ ↓ ↓

0010 0100 1001 0110

Not this is very different from converting to binary which yields:

100111000000_2

In BCD ...

0010010010010110

- BCD Code uses 4 bits to represent the 10 decimal digits {0 to 9}
- 6 BCD codes unused
- The weights of the individual positions of the bits of a BCD code are: 2^3=8, 2^2=4, 2^1=2, 2^0=1

BCD Addition

	BCD carry	1	1	
448		0100	0100	1000
+ 489		+ 0100	+ 1000	+ 1001
937	Binary sum	1001	1101	10001
	Add 6		+ 0110	+ 0110
	BCD sum		10011	10111
	BCD result	1001	0011	0111

Add each digit. If the result is greater than 9, add 6 and carry any overflow to the next digit. Repeat.

Other Decimal Codes

- 4 bits = 16 different codes
- Only 10 needed to represent the 10 decimal digits.
- Many possible codes!
- 2421 and excess-3 are self-complementing (9's complement can be obtained by inverting bits)

Four Different Binary Codes for the Decimal Digits

Decimal Digit	BCD 8421	2421	Excess-3	8, 4, −2, −1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
	1010	0101	0000	0001
Unused	1011	0110	0001	0010
bit	1100	0111	0010	0011
combi-	1101	1000	1101	1100
nations	1110	1001	1110	1101
	1111	1010	1111	1110

src: Mano's book

Gray Code

- Gray code represents decimal numbers 0 to 15 using 16 4-bit codes
- Gray codes of two adjacent decimal numbers differ by only one bit
- Example:
  - $(5)_{10} = 0111$
  - $(6)_{10} = 0101$
  - $(7)_{10} = 0100$

Number	Binary	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

- Only one bit changes with each number increment
- Not a weighted code
- Useful for interfacing to some physical systems

ASCII Character Code

- ASCII an abbreviation of "*American Standard Code for Information Interchange*"
- A 7-bit code (128 characters)
  - 94 printable, 34 non-printable (control)
    - 2x26 English letters (A,…Z, a,…z)
    - 10 decimal digits (0,1,…9)
    - 32 Special Characters such as %, *, $, … etc.
- Usually stored as a byte (8 bits)
  - The extra bit is used for other purposes

ASCII Character Code

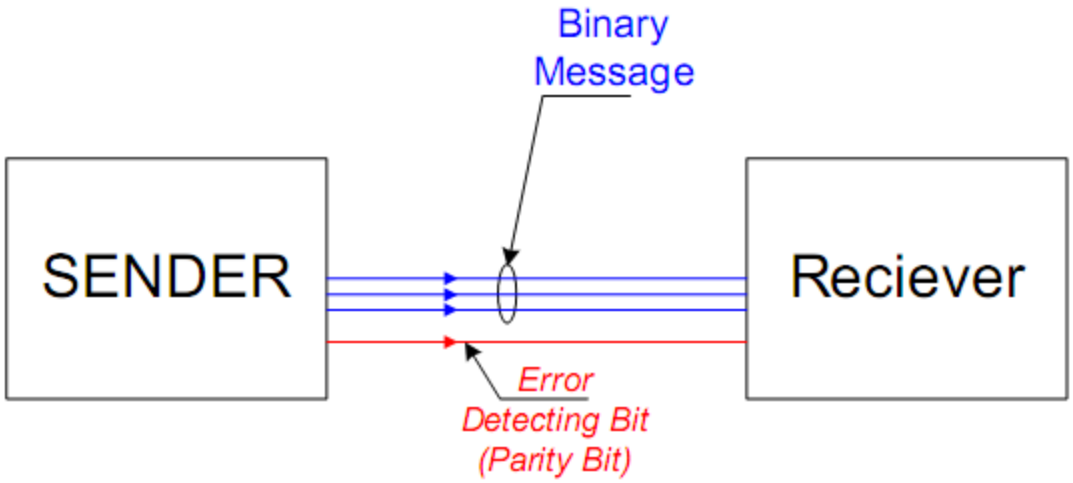
00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c
04	EOT	14	DC4	24	$	34	4	44	D	54	T	64	d
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k
0C	FF	1C	FS	2C	´	3C	<	4C	L	5C	\	6C	l
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o

ASCII Character Code

00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c
04	EOT	14	DC4	24	$	34	4	44	D	54	T	64	d
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k
0C	FF	1C	FS	2C	´	3C	<	4C	L	5C	\	6C	l
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o

capital vs small
A difference of
$(20)_{16} = 32_{10}$

Error Detecting Code



- In data communication, errors may happen
  - One code change into another code
  - How to detect errors?
- Add an extra bit called a parity bit such that
  - Number of 1's is even (even parity) or odd (odd parity)

Error Detecting Code

	Word	Even Parity	Odd Parity
ASCII A =	1000001	01000001	11000001
ASCII T =	1010100	11010100	01010100

Even Parity - number of 1 bits should be even.

Odd Parity - number of 1 bits should odd.

Parity can detect any number of odd errors: 1,3,5,... Parity is also one of the simplest ways to detect errors. Communication protocols commonly include error detection and even correction.

Conclusions

- Bits are bits
  - Modern digital devices represent everything as collections of bits
  - A computer is one such digital device
- You can encode anything with sufficient 1's and 0's
  - Binary codes (BCD, gray code)
  - Text (ASCII)
  - Sound (.wav, .mp3, ...)
  - Pictures (.jpg, .gif, .tiff)