COE 202- Digital Logic

Number Systems III

Dr. Abdulaziz Y. Barnawi

COE Department

KFUPM

Converting Decimal Integers to Binary

- Arithmetic operations:
  - Binary number system
  - Other number systems
- Binary codes
  - Binary coded decimal (BCD)
  - ASCII Code
  - Error Detecting Code

Arithmetic Operation in base-r

- Arithmetic operations with numbers in base-r follow the same rules as for decimal numbers
- Be careful !
  - – Only ***r*** allowed digits

Binary Addition

- 1 + 1 = 2, but 2 is not allowed digit in binary
- Thus, adding 1 + 1 in the binary system results in a Sum bit of 0 and a Carry bit

One bit addition:

0	**0**	**1**	**1**	← **augend** /aw-jend/
+ 0	**+ 1**	**+ 0**	**+ 1**	← **addend**
-----	------	------	------	
0	**1**	**1**	~~**2**~~	← **sum**

1 0

carry

2 doesn't exist in binary!

Binary Addition

Example:

```
          1         1 1 1
          1 1 0 0 0 0 1 1 1 1        carries
+         0 1 1 1 1 0 1 0 1 0
        -------------------------
        1 0 0 1 1 1 1 1 0 0 1        sum
```

Q: How to verify?

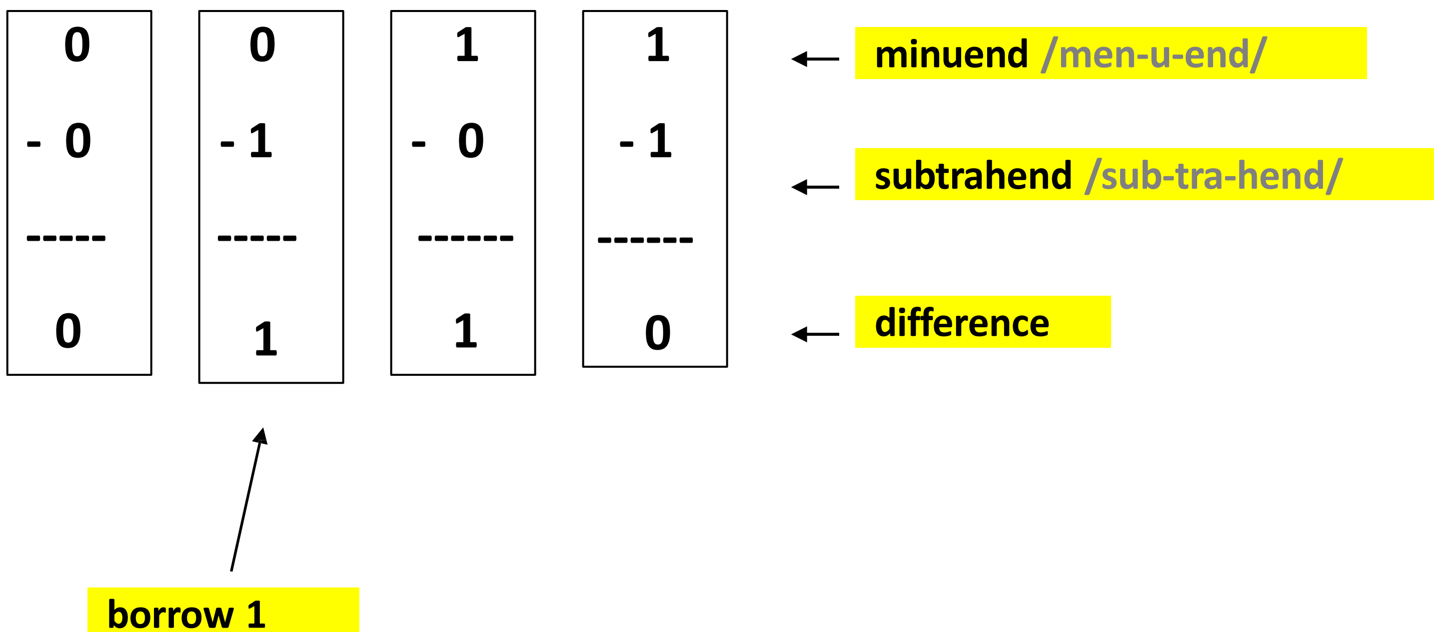
A: Convert to decimal

```
    783
+   490
-----------
   1273
```

Binary Subtraction

- The borrow digit is negative and has the weight of the next higher digit.

One bit subtraction:



Subtract 101 - 011

1

borrow

0 ~~1~~ 0 1

\- 0 1 1

----

difference

0 1 0

Larger binary numbers

1 1 1 1

~~1~~ ~~1~~ ~~0~~ ~~0~~ ~~0~~ 0 1 1 1 1

borrow

\- 0 1 1 1 1 0 1 0 1 0

----

difference

0 1 0 0 1 0 0 1 0 1

Verify In decimal,

783

\- 490

293

- In Decimal subtraction, the borrow is equal to 10.
- In Binary, the borrow is equal to 2. Therefore, a ‘1’ borrowed in binary will generate a $(10)_2$, which equals to $(2)_{10}$ in decimal

Binary Multiplication

- Binary multiplication is performed similar to decimal multiplication.
- Example: 11 * 5 = 55

Multiplicand			1	0	1	1	
Multiplier				1	0	1	x
			1	0	1	1	
		0	0	0	0		+
	1	0	1	1			+
	1	1	0	1	1	1	

Hexadecimal addition

- Rules:
  - For adding individual digits of a Hexadecimal number, a mental addition of the decimal equivalent digits makes the process easier.
  - After adding up the decimal digits, you must convert the result back to Hexadecimal, as shown in the above example.
- Example: Add $(59F)_{16}$ and $(E46)_{16}$

```
[ 1   1 ]  <-- Carry
   5 9 F
+  E 4 6
---------
 1 3 E 5
```

Carry

$F + 6 = (21)_{10} = (16 \times 1) + 5 = (15)_{16}$

$5 + E = (19)_{10} = (16 \times 1) + 3 = (13)_{16}$

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 binary coded, there is a minimum number of bits, but no maximum !

Binary Codes for Decimal Digits

- Internally, digital computers operate on binary numbers
- When interfacing to humans, digital processors, e.g. pocket calculators, communication is decimal-based
- Input is done in decimal then converted to binary for internal processing
- For output, the result has to be converted from its internal binary representation to a decimal form
- To be handled by digital processors, the decimal input (output) must be coded in binary in a digit by digit manner

Binary Codes for Decimal Digits

- For example, to input the decimal number 957, each digit of the number is individually coded and the number is stored as 1001 0101 0111.
- Thus, we need a specific code for each of the 10 decimal digits. There is a variety of such decimal binary codes.
- One commonly used code is the Binary Coded Decimal (BCD) code which corresponds to the first 10 binary representations of the decimal digits 0-9.
  - The BCD code requires 4 bits to represent the 10 decimal digits.
  - Since 4 bits may have up to 16 different binary combinations, a total of 6 combinations will be unused.
  - The position weights of the BCD code are 8, 4, 2, 1.

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:

$$\begin{array}{cccc} 2 & 4 & 9 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 0010 & 0100 & 1001 & 0110 \end{array}$$

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

$$100111000000_2$$

In BCD ...

$$0010010010010110$$

Other Decimal Codes

- 4 bits = 16 different codes
- Only 10 needed to represent the 10 decimal digits.
- Many possible codes!

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

Number Conversion versus Coding

- Converting a decimal number into binary is done by repeated division (multiplication) by 2

- Coding a decimal number into its BCD code is done by replacing each decimal digit of the number by its equivalent 4 bit BCD code.

- Example: Converting $(13)_{10}$ into binary, we get 1101, coding the same number into BCD, we obtain $(00010011)_{BCD}$.

- Exercise: Convert $(95)_{10}$ into its binary equivalent value and give its BCD code as well.

- Answer: $(1011111)_2$, and $(10010101)_{BCD}$.

ASCII Character Code

- ASCII an abbreviation of “American Standard Code for Information Interchange”
- Standard ASCII: 7-bit character codes (0 – 127)
- Extended ASCII: 8-bit character codes (0 – 255)

ASCII Codes

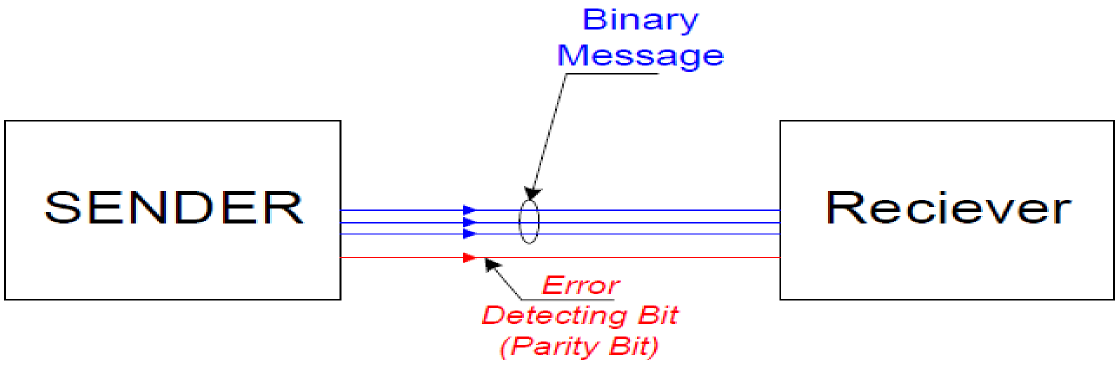
The Charcter set of the ASCII Code

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

- ASCII code for space character = 20 (hex) = 32 (decimal)
- ASCII code for ‘A' = 41 (hex) = 65 (decimal)
- ASCII code for 'a' = 61 (hex) = 97 (decimal)

Error Detection

- Binary information may be transmitted through some communication medium, e.g. using wires or wireless media.
- A corrupted bit will have its value changed from ‘0’ to ‘1’ or vice versa.
- To be able to detect errors at the receiver end, the sender sends an extra bit (parity bit) with the original binary message.



Parity Bit

- A parity bit is an extra bit included with the n-bit binary message to make the total number of 1's in this message (including the parity bit) either odd or even.
- The 8th bit in the ASCII code is used as a parity bit.
- There are two ways for error checking:
  - Even Parity: Where the 8th bit is set such that the total number of 1s in the 8-bit code word is <u>even</u>.
  - Odd Parity: The 8th bit is set such that the total number of 1s in the 8-bit code word is <u>odd</u>.

Parity Bit

Word	Even Parity	Odd Parity
1000001	01000001	11000001
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

- When performing arithmetic operations in base-r, remember allowed digits {0, .., r-1}
- 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)