King Fahd University of Petroleum & Minerals
Computer Engineering Dept

COE 200 – Fundamentals of Computer Engineering

Term 022

Dr. Ashraf S. Hasan Mahmoud

Rm 22-144

Ext. 1724

Email: ashraf@ccse.kfupm.edu.sa

Binary Addition of UNSIGEND Numbers

- Consider the following example:
  Find the summation of $(1100)_2$ and $(10001)_2$

Solution:

```
                  101100  ← Carry
      Augend       01100
      Addend      +10001
     ----------   ---------
      sum         101001
```

- Note that
  - 0+0 = 0, 0+1 = 1+0 = 1, and 1 + 1 = 0 and the carry is 1
  - If the maximum no of digits for the augend or the addend is n, then the summation has either n or n+1 digits
  - This procedure works even for non-integer binary numbers

Binary Subtraction of UNSIGEND Numbers

- Consider the following example:
  Subtract $(10010)_2$ from $(10110)_2$

Solution:

Minuend	**10110**
Subtrahend	-**10010**
---	---
Difference	**00100**

- Note that
  - $(10110)_2$ is greater than $(10010)_2$ ➔ The result is POSITIVE
  - 0-0 = 0, 1-0 = 1, and 1 - 1 = 0
  - The difference size is always less or equal to the size of the minued or the subtrahend
  - This procedure works even for non-integer binary numbers

Binary Subtraction – cont'd

- Consider the following example:
  Subtract $(10011)_2$ from $(10110)_2$

Solution:

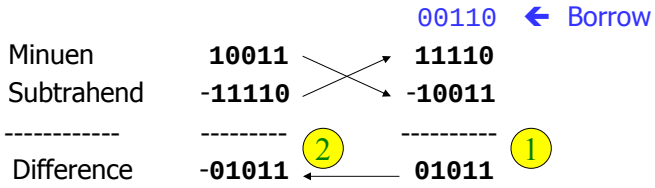
	00110 ← Borrow
Minuen	**10110**
Subtrahend	-**10011**
---	---
Difference	**00011**

- Note that
  - $(10110)_2$ is greater than $(10011)_2$ ➔ result is positive
  - 0-1= 1, and the borrow from next significant digit is 1
  - This procedure works even for non-integer binary numbers

Binary Subtraction – cont'd

- Consider the following example:
  Subtract $(11110)_2$ from $(10011)_2$

Solution:



- Note that
  - $(10011)_2$ is smaller than $(11110)_2$ ➔ result is negative
  - This procedure works even for non-integer binary numbers

Binary Multiplication of UNSIGEND Numbers

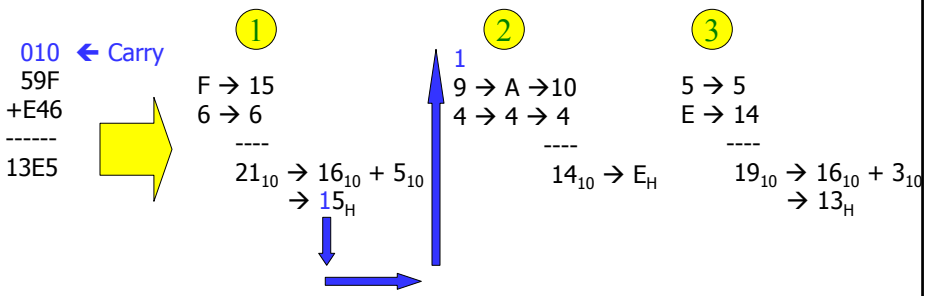
- Consider the following example:
  Multiply $(1011)_2$ by $(101)_2$

Solution:

```
Multiplicand      1011
Multiplier      X 101
------------    ------
                 1011
                0000
               1011
              -------
Product       110111
```

Sums and Products in Base r (Unsigned Numbers)

- For sums and Products in base-r (r > 2) systems
  - Memorize tables for sums and products
  - Convert to Dec ➔ perform operation ➔ convert back to base-r
- **Example**: Find the summation of $(59F)_{16}$ and $(E46)_{16}$?



- This procedure is used for any base-r

Sums and Products in Base-r – cont'd

- **Example**: Find the multiplication of $(762)_8$ and $(45)_8$?
- **Solution**:

3310 ← Carry (for 4)
4310 ← Carry (for 5)
762
X 45

4672
3710

43772

Octal	Decimal	Octal
5X2	= 10 → 8 + 2	= 12
5X6+1	= 31 →24 + 7	= 37
5X7+3	= 38 →32 + 6	= 46
4X2	= 8 → 8 + 0	= 10
4X6+1	= 25 → 24 + 1	= 31
4X7+3	= 24+7	= 37

Therefore, product = $(43772)_8$

Decimal Codes

- There are 2^n **DISTINCT** n-bit binary codes (group of n bits)
  - n bits can count 2^n numbers
- For us, humans, it is more natural to deal with decimal digits rather than binary digits
- 10 different digits ➔ we can use 4 bits to represent any digit
  - 3 bits count 8 numbers
  - 4 bits count 16 numbers ➔ to represent 10 digits we need 4 bits at least

Binary Coded Decimal (BCD)

- Let the decimal digits be coded as show in table
- Then we can write numbers as

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

$(396)_{10} = (0011\ 1001\ 0110)_{BCD}$

Although we are using the equal sign – but they are not equal in the mathematical sense; this is **JUST a code**

Since 3 ➔ 0011, 9 = 1001, 6 = 0110

Note that $(396)_{10} = (110001100)_2 \neq (0011\ 1001\ 0110)_{BCD}$

BCD Addition – Example 1

- Consider:

000 ← Carry
241
+105

346

Addition in the Decimal Domain

0010 → Carry
BCD for 1 = 0001
BCD for 5 = 0101

0110 → BCD for 6

0000 → Carry
BCD for 4 = 0100
BCD for 0 = 0000

0100 → BCD for 4

0000 → Carry
BCD for 2 = 0010
BCD for 1 = 0001

0011 → BCD for 3

Addition in the Decimal Domain

Hence, we can add BCD codes to obtain the correct decimal result. Is true always?

BCD Addition – Example 2

- Consider:

110 ← Carry
448
+489

937

Addition in the Decimal Domain

0010 → Carry
BCD for 8 = 1000
BCD for 9 = 1001
---
1 0001 → > 9 → Need a correction step
+0110 (add 6)
---
1 0111 → (BCD for 7)

0001 → Carry
BCD for 4 = 0100
BCD for 8 = 1000

1101 → > 9 → Need a correction step
+0110 (add 6)
---
1 0011 → (BCD for 3)

0001 → Carry
BCD for 4 = 0100
BCD for 4 = 0100

1001 → BCD for 9

Addition in the Decimal Domain

BCD Addition – Summary

- BCD codes: decimal digits are assigned 4 bit codes
- We can perform additions using the BCD digits
  - If the result of adding two BCD digits is greater than 9, a correction step is required in order produce the correct BCD digit
  - To correct: add 6
  - If a carry is produced → move it to next BCD digits addition

Alphanumeric Codes

- We have
  - 10 decimal digits
  - 26 X 2 (English) letters: capital and small case
  - Some special characters {; , . : + - etc}
- If we assign each character of these a binary code, then computers can exchange alphanumeric information (letters, numbers, etc) by exchanging binary digits
- One binary code is the American Standard Code for Information Interchange (ASCII)

ASCII

- A 7-bits code ➔ 128 distinct codes
  - 96 printable characters (26 upper case letter, 26 lower case letters, 10 decimal digits, 34 non-alphanumeric characters)
  - 32 non-printable character
    - Formatting effectors (CR, BS, …)
    - Info separators (RS, FS, …)
    - Communication control (STX, ETX, …)
- Computers typically use words sizes that are multiples of 2
  - Usually 8 bits are used for the ASCII code with the 8th (left most bit) bit set to zero, OR
  - The ASCII code is extended ➔Extended ASCII (platform dependant)
- A good reference about ASCII and Extended ASCII is found at http://www.cplusplus.com/doc/papers/ascii.html

ASCII – cont'd

- A 7-bits code ➔ 128 distinct codes
- The American Standard Code for Information Interchange (ASCII) uses seven binary digits to represent 128 characters as shown in the table.

00	NUL	01	SOH	02	STX	03	ETX	04	EOT	05	ENQ	06	ACK	07	BEL
08	BS	09	HT	0A	NL	0B	VT	0C	NP	0D	CR	0E	SO	0F	SI
10	DLE	11	DC1	12	DC2	13	DC3	14	DC4	15	NAK	16	SYN	17	ETB
18	CAN	19	EM	1A	SUB	1B	ESC	1C	FS	1D	GS	1E	RS	1F	US
20	SP	21	!	22	"	23	#	24	$	25	%	26	&	27	'
28	(	29	)	2A	*	2B	+	2C	,	2D	-	2E	.	2F	/
30	0	31	1	32	2	33	3	34	4	35	5	36	6	37	7
38	8	39	9	3A	:	3B	;	3C	<	3D	=	3E	>	3F	?
40	@	41	A	42	B	43	C	44	D	45	E	46	F	47	G
48	H	49	I	4A	J	4B	K	4C	L	4D	M	4E	N	4F	O
50	P	51	Q	52	R	53	S	54	T	55	U	56	V	57	W
58	X	59	Y	5A	Z	5B	[	5C	\	5D	]	5E	^	5F	_
60	`	61	a	62	b	63	c	64	d	65	e	66	f	67	g
68	h	69	i	6A	j	6B	k	6C	l	6D	m	6E	n	6F	o
70	p	71	q	72	r	73	s	74	t	75	u	76	v	77	w
78	x	79	y	7A	z	7B	{	7C	|	7D	}	7E	~	7F	DEL

Unicode

- Unicode describes a 16-bit standard code for representing symbols and ideographs for the world's languages.

First 256 Codes for Unicode[a]

	Control		ASCII						Control		Latin 1					
	000	001	002	003	004	005	006	007	008	009	00A	00B	00C	00D	00E	00F
0	CTRL	CTRL	SPACE	0	@	P	`	p	CTRL	CTRL	NB SP	°	À	Ð	à	ð
1	CTRL	CTRL	!	1	A	Q	a	q	CTRL	CTRL	¡	±	Á	Ñ	á	ñ
2	CTRL	CTRL	"	2	B	R	b	r	CTRL	CTRL	¢	²	Â	Ò	â	ò
3	CTRL	CTRL	#	3	C	S	c	s	CTRL	CTRL	£	³	Ã	Ó	ã	ó
4	CTRL	CTRL	$	4	D	T	d	t	CTRL	CTRL	¤	´	Ä	Ô	ä	ô
5	CTRL	CTRL	%	5	E	U	e	u	CTRL	CTRL	¥	µ	Å	Õ	å	õ
6	CTRL	CTRL	&	6	F	V	f	v	CTRL	CTRL	¦	¶	Æ	Ö	æ	ö
7	CTRL	CTRL	'	7	G	W	g	w	CTRL	CTRL	§	·	Ç	×	ç	÷
8	CTRL	CTRL	(	8	H	X	h	x	CTRL	CTRL	¨	¸	È	Ø	è	ø
9	CTRL	CTRL	)	9	I	Y	i	y	CTRL	CTRL	©	¹	É	Ù	é	ù
A	CTRL	CTRL	*	:	J	Z	j	z	CTRL	CTRL	ª	º	Ê	Ú	ê	ú
B	CTRL	CTRL	+	;	K	[	k	{	CTRL	CTRL	«	»	Ë	Û	ë	û
C	CTRL	CTRL	,	<	L	\	l	|	CTRL	CTRL	¬	¼	Ì	Ü	ì	ü
D	CTRL	CTRL	-	=	M	]	m	}	CTRL	CTRL	-	½	Í	Ý	í	ý
E	CTRL	CTRL	.	>	N	^	n	~	CTRL	CTRL	®	¾	Î	Þ	î	þ
F	CTRL	CTRL	/	?	O	_	o	CTRL	CTRL	CTRL	¯	¿	Ï	ß	ï	ÿ

[a] Unicode, Inc., The Unicode Standard: Worldwide Character Encoding, Version 1.0, Volume 1, © 1990, 1991 by Unicode, Inc. Reprinted by permission of Addison-Wesley Publishing Company, Inc.

Problems of Interest

- Problem List:

- Homework: Chapter 1, pages 24-26**:** 2, 3, 8, 12, 14, 16, 19, 24, 26

  Due date: Monday March 17, 2003 (in class)