

Codes and number systems

*Introduction to Computer
Yung-Yu Chuang*

with slides by Nisan & Schocken (www.nand2tetris.org) and Harris & Harris (DDCA)

Coding

- Assume that you want to communicate with your friend with a flashlight in a night, what will you do?



light painting?
What's the problem?

Solution #1



- A: 1 blink
- B: 2 blinks
- C: 3 blinks
- :
- Z: 26 blinks

What's the problem?

- How are you? = 131 blinks

Solution #2: Morse code



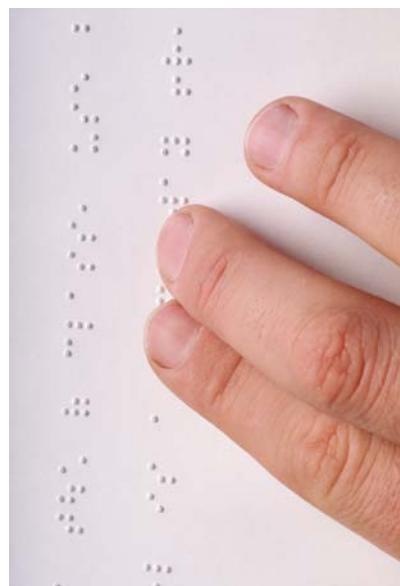
A	—	J	----	S	---
B	—...	K	—--	T	-
C	---.	L	...-	U	---
D	---	M	--	V	---
E	.	N	--	W	---
F	...-	O	----	X	---
G	---	P	...-	Y	---
H	Q	---	Z	---
I	..	R	---		

● ● ● ● ● — ● ● ● — ● ● — — —

Hello



Braille



4	5	6
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1	2	3



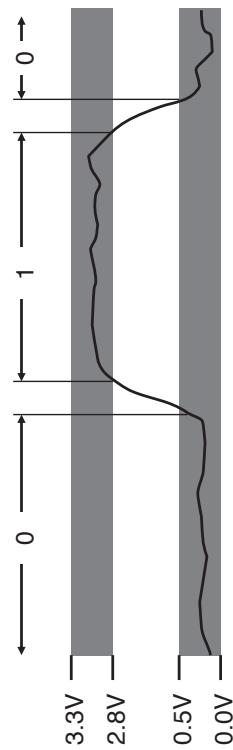
What's common in these codes?



Binary representations

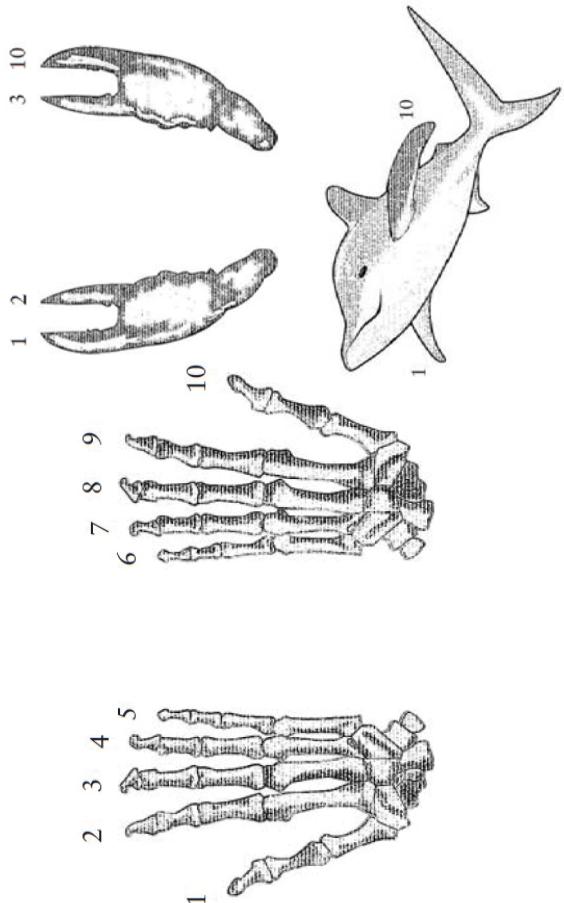
- They are both binary codes.

- Electronic Implementation
 - Easy to store with bistable elements
 - Reliably transmitted on noisy and inaccurate wires



Number systems

Number Systems



- Decimal numbers

1's column
10's column
100's column
1000's column

$$5374_{10} =$$

- Binary numbers

1's column
2's column
4's column
8's column

$$1101_2 =$$

FROM ZERO TO ONE

Chapter 1 <10>



Number Systems

- Decimal numbers

1's column
10's column
100's column
1000's column

$$5374_{10} = 5 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 4 \times 10^0$$

five three seven four
thousands hundreds tens ones

- Binary numbers

1's column
2's column
4's column
8's column

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$$

one one no one
eight four two one

© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <11>



Binary numbers

- Digits are 1 and 0
(a binary digit is called a bit)

1 = true

0 = false

- MSB -most significant bit
- LSB -least significant bit

- Bit numbering:

MSB	1 0 1 1 0 0 1 0 1 0 0 1 1 1 0 0	LSB
	15	0

- A bit string could have different interpretations

© Digital Design and Computer Architecture, 2nd Edition, 2012



Chapter 1 <10>

© Digital Design and Computer Architecture, 2nd Edition, 2012



Powers of Two

FROM ZERO TO ONE

- $2^0 =$
- $2^1 =$
- $2^2 =$
- $2^3 =$
- $2^4 =$
- $2^5 =$
- $2^6 =$
- $2^7 =$

- $2^8 =$
- $2^9 =$
- $2^{10} =$
- $2^{11} =$
- $2^{12} =$
- $2^{13} =$
- $2^{14} =$
- $2^{15} =$

© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <13>

© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <14>



Powers of Two

FROM ZERO TO ONE

- $2^0 = 1$
- $2^1 = 2$
- $2^2 = 4$
- $2^3 = 8$
- $2^4 = 16$
- $2^5 = 32$
- $2^6 = 64$
- $2^7 = 128$
- $2^8 = 256$
- $2^9 = 512$
- $2^{10} = 1024$
- $2^{11} = 2048$
- $2^{12} = 4096$
- $2^{13} = 8192$
- $2^{14} = 16384$
- $2^{15} = 32768$
- Handy to memorize up to 2^9

© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <14>

Unsigned binary integers

- Each digit (bit) is either 1 or 0
- Each bit represents a power of 2:

1	1	1	1	1	1	1	1
2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰

Table 1-3 Binary Bit Position Values.

2 ⁿ	Decimal Value	2 ⁿ	Decimal Value
2 ⁰	1	2 ⁸	256
2 ¹	2	2 ⁹	512
2 ²	4	2 ¹⁰	1024
2 ³	8	2 ¹¹	2048
2 ⁴	16	2 ¹²	4096
2 ⁵	32	2 ¹³	8192
2 ⁶	64	2 ¹⁴	16384
2 ⁷	128	2 ¹⁵	32768

Every binary number is a sum of powers of 2



Translating binary to decimal

- Weighted positional notation shows how to calculate the decimal value of each binary bit:

$$dec = (D_{n-1} \times 2^{n-1}) + (D_{n-2} \times 2^{n-2}) + \dots + (D_1 \times 2^1) + (D_0 \times 2^0)$$

D = binary digit

binary 00001001 = decimal 9:

$$(1 \times 2^3) + (1 \times 2^0) = 9$$

Translating unsigned decimal to binary

Number Conversion

- Repeatedly divide the decimal integer by 2. Each remainder is a binary digit in the translated value:
 - Convert 10011_2 to decimal

Division	Quotient	Remainder
$37 / 2$	18	1
$18 / 2$	9	0
$9 / 2$	4	1
$4 / 2$	2	0
$2 / 2$	1	0
$1 / 2$	0	1

$$37 = 100101$$

- Binary to decimal conversion:
 - Convert 47_{10} to binary

Division	Quotient	Remainder
$47 / 2$	23	1
$23 / 2$	11	1
$11 / 2$	5	1
$5 / 2$	2	1
$2 / 2$	1	0
$1 / 2$	0	1

- Decimal to binary conversion:
 - Convert 47_{10} to binary



Number Conversion

- Binary to decimal conversion:

- Convert 10011_2 to decimal
 - $16 \times 1 + 8 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 1 = 19_{10}$

- Decimal to binary conversion:
 - Convert 47_{10} to binary
 - $32 \times 1 + 16 \times 0 + 8 \times 1 + 4 \times 1 + 2 \times 1 + 1 \times 1 = 101111_2$

Binary Values and Range

- N-digit decimal number

- How many values?
- Range?
- Example: 3-digit decimal number:

- 3-digit decimal number:
 - Range:
 - Example:

- N-bit binary number
 - How many values?
 - Range:
 - Example: 3-digit binary number:



Binary Values and Range

- N -digit decimal number
 - How many values? 10^N
 - Range? **[0, $10^N - 1$]**
 - Example: 3-digit decimal number:
 - $10^3 = 1000$ possible values
 - Range: [0, 999]
 - N -bit binary number
 - How many values? 2^N
 - Range: **[0, $2^N - 1$]**
 - Example: 3-digit binary number:
 - $2^3 = 8$ possible values
 - Range: [0, 7] = [000₂, to 111₂]

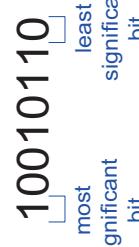
© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <21>



ELSEVIER

Bits, Bytes, Nibbles...

- Bits
 -  100010110
 -  10010110
- Bytes & Nibbles

FROM ZERO TO ONE

Large Powers of Two

- $2^{10} = 1$ kilo ≈ 1000 (1024)
- $2^{20} = 1$ mega ≈ 1 million (1,048,576)
- $2^{30} = 1$ giga ≈ 1 billion (1,073,741,824)

Practice: What is the largest unsigned integer that may be stored in 20 bits?

FROM ZERO TO ONE

Integer storage sizes

Standard sizes:	byte	8
	word	16
	doubleword	32
	quadword	64

Table 1-4 Ranges of Unsigned Integers.

Storage Type	Range (low-high)	Powers of 2
Unsigned byte	0 to 255	0 to ($2^8 - 1$)
Unsigned word	0 to 65,535	0 to ($2^{16} - 1$)
Unsigned doubleword	0 to 4,294,967,295	0 to ($2^{32} - 1$)
Unsigned quadword	0 to 18,446,744,073,709,551,615	0 to ($2^{64} - 1$)

Practice: What is the largest unsigned integer that may be stored in 20 bits?

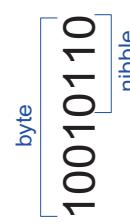
FROM ZERO TO ONE

Large Powers of Two

- $2^{10} = 1$ kilo ≈ 1000 (1024)
- $2^{20} = 1$ mega ≈ 1 million (1,048,576)
- $2^{30} = 1$ giga ≈ 1 billion (1,073,741,824)

FROM ZERO TO ONE

Bytes & Nibbles

- Bytes
 -  CEBF9AD7
 -  10010110
- Bytes & Nibbles

© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <23>

© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <24>



ELSEVIER



ELSEVIER

Estimating Powers of Two

- What is the value of 2^{24} ?
 - How many values can a 32-bit variable represent?
 $-2^2 \times 2^{20} \approx 16 \text{ million}$

Estimating Powers of Two

- What is the value of 2^{24} ?
 - How many values can a 32-bit variable represent?
 $-2^2 \times 2^{20} \approx 16 \text{ million}$

Large measurements

- Kilobyte (KB), 2^{10} bytes
- Megabyte (MB), 2^{20} bytes
- Gigabyte (GB), 2^{30} bytes
- Terabyte (TB), 2^{40} bytes
- Petabyte
- Exabyte
- Zettabyte
- Yottabyte

FROM ZERO TO ONE

Hexadecimal Numbers

Hex Digit	Decimal Equivalent	Binary Equivalent
0	0	
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
A	10	
B	11	
C	12	
D	13	
E	14	
F	15	

Hexadecimal Numbers



Hexadecimal Numbers

Hex Digit	Decimal Equivalent	Binary Equivalent
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Digital Design and Computer Architecture, 2nd Edition, 2012 Chapter 1 <29>

Translating binary to hexadecimal

- Each hexadecimal digit corresponds to 4 binary bits.
 - Example: Translate the binary integer 000101101010011110010100 to hexadecimal:



Converting hexadecimal to decimal



Chapter 1 <30>

© Digital Design and Computer Architecture, 2nd Edition - 2012

Chapter 1 <29>

- Base 16
- Shorthand for binary



- Multiply each digit by its corresponding power of 16:

$$\text{dec} = (\mathbf{D}_3 \times 16^3) + (\mathbf{D}_2 \times 16^2) + (\mathbf{D}_1 \times 16^1) + (\mathbf{D}_0 \times 16^0)$$

- Hex 1234 equals $(1 \times 16^3) + (2 \times 16^2) + (3 \times 16^1) + (4 \times 16^0)$, or decimal 4,660.
 - Hex 3BA4 equals $(3 \times 16^3) + (11 \times 16^2) + (10 \times 16^1) + (4 \times 16^0)$, or decimal 15,268.



Hexadecimal to Binary Conversion

- Hexadecimal to binary conversion:

- Convert $4AF_{16}$ (also written $0x4AF$) to binary

- Convert $4AF_{16}$ (also written $0x4AF$) to binary

- $0100\ 1010\ 1111_2$

- Hexadecimal to decimal conversion:

- Convert $4AF_{16}$ to decimal

- $16^2 \times 4 + 16^1 \times 10 + 16^0 \times 15 = 1199_{10}$

Hexadecimal to Binary Conversion

- Hexadecimal to binary conversion:

- Convert $4AF_{16}$ (also written $0x4AF$) to binary

- $0100\ 1010\ 1111_2$

- Hexadecimal to decimal conversion:

- Convert $4AF_{16}$ to decimal

- $16^2 \times 4 + 16^1 \times 10 + 16^0 \times 15 = 1199_{10}$

	© Digital Design and Computer Architecture, 2 nd Edition, 2012
Chapter 1 <33>	

Powers of 16

Used when calculating hexadecimal values up to
8 digits long:

16^n	Decimal Value	16^n	Decimal Value
16^0	1	16^4	65,536
16^1	16	16^5	1,048,576
16^2	256	16^6	16,777,216
16^3	4096	16^7	268,435,456



Converting decimal to hexadecimal

Division	Quotient	Remainder
$422 / 16$	26	6
$26 / 16$	1	A
$1 / 16$	0	1

decimal $422 = 1A6$ hexadecimal



Chapter 1 <34>

Addition

- Decimal
$$\begin{array}{r} 11 \leftarrow \text{carries} \\ 3734 \\ + 5168 \\ \hline 8902 \end{array}$$

- Binary
$$\begin{array}{r} 11 \leftarrow \text{carries} \\ 1011 \\ + 0011 \\ \hline 1110 \end{array}$$

© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <37>

© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <38>

ELSEVIER



Binary Addition Examples

- Add the following 4-bit binary numbers

$$\begin{array}{r} 1001 \\ + 0101 \\ \hline \end{array}$$

$$\begin{array}{r} 1011 \\ + 0110 \\ \hline \end{array}$$

© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <38>

ELSEVIER



Binary Addition Examples

- Add the following 4-bit binary numbers

$$\begin{array}{r} 1 \\ 1001 \\ + 0101 \\ \hline 1110 \end{array}$$

- Add the following 4-bit binary numbers

$$\begin{array}{r} 111 \\ 1011 \\ + 0110 \\ \hline 10001 \end{array}$$

Overflow!

© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <39>

ELSEVIER



Overflow

- Digital systems operate on a **fixed number of bits**
- Overflow: when result is too big to fit in the available number of bits
- See previous example of $11 + 6$

© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <40>

ELSEVIER



Hexadecimal addition



Signed Binary Numbers

Divide the sum of two digits by the number base (16). The quotient becomes the carry value, and the remainder is the sum digit.

$$\begin{array}{r} 36 & 28 & 1 & 1 \\ 42 & 45 & 28 & 6A \\ \hline 78 & 6D & 80 & B5 \end{array}$$

Important skill: Programmers frequently add and subtract the addresses of variables and instructions.



Chapter 1 <42>

© Digital Design and Computer Architecture, 2nd Edition, 2012

Signed integers

The highest bit indicates the sign. 1 = negative, 0 = positive



- Sign/Magnitude Numbers
- Two's Complement Numbers

$$A = (-1)^{a_{n-1}} \sum_{i=0}^{n-2} a_i 2^i$$

- Example, 4-bit sign/mag representations of ± 6 :

$$+6 =$$

$$-6 =$$

- Range of an N -bit sign/magnitude number:

If the highest digit of a hexadecimal integer is > 7 , the value is negative. Examples: 8A, C5, A2, 9D



Chapter 1 <44>

© Digital Design and Computer Architecture, 2nd Edition, 2012

Sign/Magnitude Numbers

- 1 sign bit, $N-1$ magnitude bits
 - Sign bit is the most significant (left-most) bit
 - Positive number: sign bit = 0 $A : \{a_{N-1}, a_{N-2}, \dots, a_2, a_1, a_0\}$
 - Negative number: sign bit = 1 $A = (-1)^{a_{n-1}} \sum_{i=0}^{n-2} a_i 2^i$
 - Example, 4-bit sign/mag representations of ± 6 :
 - $+6 = \textbf{0110}$
 - $-6 = \textbf{1110}$
 - Range of an N -bit sign/magnitude number:
 $[-(2^{N-1}-1), 2^{N-1}-1]$

© Digital Design and Computer Architecture 2nd Edition 2013

Digital Design and Computer Architecture 3rd Edition 2013 Chapter 1 <46>

© Digital Design and Computer Architecture 2nd Edition 2013

Chapter 1 <45>

Two's Complement Numbers

- Don't have same problems as sign/magnitude numbers:
 - Addition works
 - Single representation for 0

Digital Design and Computer Architecture 2nd Edition 2012 Chapter 1 <46>

Chapter 1 <45>

80

Two's complement notation

stance.

- Complement (reverse) each bit
 - Add 1

Starting value	00000001
Step 1: reverse the bits	11111110
Step 2: add 1 to the value from Step 1	$ \begin{array}{r} 11111110 \\ +00000001 \\ \hline \end{array} $
Sum: two's complement representation	11111111

REVIEWS

Chapter 1 <47>

© Digital Design and Computer Architecture 2nd Edition 2013

Sign/Magnitude Numbers

- Problems:

 - Addition doesn't work, for example $-6 + 6$:
$$\begin{array}{r} 1110 \\ + 0110 \\ \hline 10100 \text{ (wrong!)} \end{array}$$
 - Two representations of 0 (± 0):
1000
0000

“Taking the Two’s Complement”

- Flip the sign of a two’s complement number
- Method:
 1. Invert the bits
 2. Add 1
- Example: Flip the sign of $3_{10} = 0011_2$

$$\begin{array}{r} \text{1. } 1100 \\ \text{2. } + \quad 1 \\ \hline \text{1101} = -3_{10} \end{array}$$

“Taking the Two’s Complement”

- Flip the sign of a two’s complement number
- Method:
 1. Invert the bits
 2. Add 1
- Example: Flip the sign of $3_{10} = 0011_2$

$$\begin{array}{r} \text{1. } 1100 \\ \text{2. } + \quad 1 \\ \hline \text{1101} = -3_{10} \end{array}$$

Two’s Complement Examples

- Take the two’s complement of $6_{10} = 0110_2$
- What is the decimal value of 1001_2 ?

FROM ZERO TO ONE

Two’s Complement Examples

- Take the two’s complement of $6_{10} = 0110_2$
- What is the decimal value of the two’s complement number 1001_2 ?

$$1010_2 = -6_{10}$$

$$\begin{array}{r} \text{1. } 0110 \\ \text{2. } + \quad 1 \\ \hline 0111_2 = 7_{10}, \text{ so } 1001_2 = -7_{10} \end{array}$$

Binary subtraction



- When subtracting $A - B$, convert B to its two's complement

- Add A to $(-B)$

$$\begin{array}{r} 01010 \\ - 01011 \\ \hline \end{array} \quad \longrightarrow \quad \begin{array}{r} 01010 \\ 10101 \\ \hline 11111 \end{array}$$

Advantages for 2's complement:

- No two 0's
- Sign bit
- Remove the need for separate circuits for add and sub

Two's Complement Addition

- Add 6 + (-6) using two's complement numbers
- Add -2 + 3 using two's complement numbers

$$\begin{array}{r} 0110 \\ + 1010 \\ \hline \end{array}$$

$$\begin{array}{r} 1110 \\ + 0011 \\ \hline \end{array}$$



Chapter 1 <54>

© Digital Design and Computer Architecture, 2nd Edition, 2012

Two's Complement Addition

- Add 6 + (-6) using two's complement numbers
- Add -2 + 3 using two's complement numbers

$$\begin{array}{r} 111 \\ 0110 \\ + 1010 \\ \hline 10000 \end{array}$$

$$\begin{array}{r} 111 \\ 1110 \\ + 0011 \\ \hline 10001 \end{array}$$



Chapter 1 <55>

© Digital Design and Computer Architecture, 2nd Edition, 2012

Two's Complement

- In 2's complement, $2^N - a$ is used to represent $-a$ since $2^N - a \equiv -a \pmod{2^N}$
- $\bar{b} = 1 - b$
- $\bar{a} = 111\dots1 - a = (2^N - 1) - a$
- $\bar{a} + 1 = 2^N - a$



Chapter 1 <56>

© Digital Design and Computer Architecture, 2nd Edition, 2012

Increasing Bit Width

- Extend number from N to M bits ($M > N$):

- Sign-extension
- Zero-extension

Copyright © 2012 Elsevier
© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <57>

Chapter 1 <58>

Chapter 1 <59>



ELSEVIER

Sign-Extension

- Sign bit copied to msb's

- Number value is same

- Example 1:

- 4-bit representation of 3 = **0011**
- 8-bit sign-extended value: **00000011**

- Example 2:

- 4-bit representation of -5 = **1011**
- 8-bit sign-extended value: **11111011**

Zero-Extension

- Zeros copied to msb's
- Value changes for negative numbers

- Example 1:

- 4-bit value = $0011_2 = 3_{10}$
- 8-bit zero-extended value: **00000011** = 3_{10}

- Example 2:

- 4-bit value = $1011 = -5_{10}$
- 8-bit zero-extended value: **00001011** = 11_{10}

Copyright © 2012 Elsevier
© Digital Design and Computer Architecture, 2nd Edition, 2012

Chapter 1 <59>

Chapter 1 <60>

Chapter 1 <61>



ELSEVIER

Chapter 1 <62>

Chapter 1 <63>

Chapter 1 <64>

Chapter 1 <65>

Chapter 1 <66>

Chapter 1 <67>

Chapter 1 <68>

Chapter 1 <69>

Chapter 1 <70>

Chapter 1 <71>

Chapter 1 <72>

Chapter 1 <73>

Chapter 1 <74>

Chapter 1 <75>

Chapter 1 <76>

Chapter 1 <77>

Chapter 1 <78>

Chapter 1 <79>

Chapter 1 <80>

Chapter 1 <81>

Chapter 1 <82>

Chapter 1 <83>

Chapter 1 <84>

Chapter 1 <85>

Chapter 1 <86>

Chapter 1 <87>

Chapter 1 <88>

Chapter 1 <89>

Chapter 1 <90>

Chapter 1 <91>

Chapter 1 <92>

Chapter 1 <93>

Chapter 1 <94>

Chapter 1 <95>

Chapter 1 <96>

Chapter 1 <97>

Chapter 1 <98>

Chapter 1 <99>

Chapter 1 <100>

Chapter 1 <101>

Chapter 1 <102>

Chapter 1 <103>

Chapter 1 <104>

Chapter 1 <105>

Chapter 1 <106>

Chapter 1 <107>

Chapter 1 <108>

Chapter 1 <109>

Chapter 1 <110>

Chapter 1 <111>

Chapter 1 <112>

Chapter 1 <113>

Chapter 1 <114>

Chapter 1 <115>

Chapter 1 <116>

Chapter 1 <117>

Chapter 1 <118>

Chapter 1 <119>

Chapter 1 <120>

Chapter 1 <121>

Chapter 1 <122>

Chapter 1 <123>

Chapter 1 <124>

Chapter 1 <125>

Chapter 1 <126>

Chapter 1 <127>

Chapter 1 <128>

Chapter 1 <129>

Chapter 1 <130>

Chapter 1 <131>

Chapter 1 <132>

Chapter 1 <133>

Chapter 1 <134>

Chapter 1 <135>

Chapter 1 <136>

Chapter 1 <137>

Chapter 1 <138>

Chapter 1 <139>

Chapter 1 <140>

Chapter 1 <141>

Chapter 1 <142>

Chapter 1 <143>

Chapter 1 <144>

Chapter 1 <145>

Chapter 1 <146>

Chapter 1 <147>

Chapter 1 <148>

Chapter 1 <149>

Chapter 1 <150>

Chapter 1 <151>

Chapter 1 <152>

Chapter 1 <153>

Chapter 1 <154>

Chapter 1 <155>

Chapter 1 <156>

Chapter 1 <157>

Chapter 1 <158>

Chapter 1 <159>

Chapter 1 <160>

Chapter 1 <161>

Chapter 1 <162>

Chapter 1 <163>

Chapter 1 <164>

Chapter 1 <165>

Chapter 1 <166>

Chapter 1 <167>

Chapter 1 <168>

Chapter 1 <169>

Chapter 1 <170>

Chapter 1 <171>

Chapter 1 <172>

Chapter 1 <173>

Chapter 1 <174>

Chapter 1 <175>

Chapter 1 <176>

Chapter 1 <177>

Chapter 1 <178>

Chapter 1 <179>

Chapter 1 <180>

Chapter 1 <181>

Chapter 1 <182>

Chapter 1 <183>

Chapter 1 <184>

Chapter 1 <185>

Chapter 1 <186>

Chapter 1 <187>

Chapter 1 <188>

Chapter 1 <189>

Chapter 1 <190>

Chapter 1 <191>

Chapter 1 <192>

Chapter 1 <193>

Chapter 1 <194>

Chapter 1 <195>

Chapter 1 <196>

Chapter 1 <197>

Chapter 1 <198>

Chapter 1 <199>

Chapter 1 <200>

Chapter 1 <201>

Chapter 1 <202>

Chapter 1 <203>

Chapter 1 <204>

Chapter 1 <205>

Chapter 1 <206>

Chapter 1 <207>

Chapter 1 <208>

Chapter 1 <209>

Chapter 1 <210>

Chapter 1 <211>

Chapter 1 <212>

Chapter 1 <213>

Chapter 1 <214>

Chapter 1 <215>

Chapter 1 <216>

Chapter 1 <217>

Chapter 1 <218>

Chapter 1 <219>

Chapter 1 <220>

Chapter 1 <221>

Chapter 1 <222>

Chapter 1 <223>

Chapter 1 <224>

Chapter 1 <225>

Chapter 1 <226>

Chapter 1 <227>

Chapter 1 <228>

Chapter 1 <229>

Chapter 1 <230>

Chapter 1 <231>

Chapter 1 <232>

Chapter 1 <233>

Chapter 1 <234>

Chapter 1 <235>

Chapter 1 <236>

Chapter 1 <237>

Chapter 1 <238>

Chapter 1 <239>

Chapter 1 <240>

Chapter 1 <241>

Chapter 1 <242>

Chapter 1 <243>

Chapter 1 <244>

Chapter 1 <245>

Chapter 1 <246>

Chapter 1 <247>

Chapter 1 <248>

Chapter 1 <249>

Chapter 1 <250>

Chapter 1 <251>

Chapter 1 <252>

Chapter 1 <253>

Chapter 1 <254>

Chapter 1 <255>

Chapter 1 <256>

Chapter 1 <257>

Chapter 1 <258>

Chapter 1 <259>

Chapter 1 <260>

Chapter 1 <261>

Chapter 1 <262>

Chapter 1 <263>

Chapter 1 <264>

Chapter 1 <265>

Chapter 1 <266>

Chapter 1 <267>

Chapter 1 <268>

Chapter 1 <269>

Chapter 1 <270>

Chapter 1 <271>

Chapter 1 <272>

Chapter 1 <273>

Chapter 1 <274>

Chapter 1 <275>

Chapter 1 <276>

Chapter 1 <277>

Chapter 1 <278>

Chapter 1 <279>

Chapter 1 <280>

Chapter 1 <281>

Chapter 1 <282>

Chapter 1 <283>

Chapter 1 <284>

Chapter 1 <285>

Chapter 1 <286>

Chapter 1 <287>

Chapter 1 <288>

Chapter 1 <289>

Chapter 1 <290>

Chapter 1 <291>

Chapter 1 <292>

Chapter 1 <293>

Chapter 1 <294>

Chapter 1 <295>

Chapter 1 <296>

Chapter 1 <297>

Chapter 1 <298>

Chapter 1 <299>

<p

Ranges of signed integers

Character

The highest bit is reserved for the sign. This limits the range:

- Character sets
 - Standard ASCII (0 – 127)
 - Extended ASCII (0 – 255)
 - ANSI (0 – 255)
 - Unicode (0 – 65,535)
 - Null-terminated String
 - Array of characters followed by a null character (0)
 - Using the ASCII table
 - back inside cover of book

Storage Type	Range (low–high)	Powers of 2
Signed byte	-128 to +127	-2^7 to $(2^7 - 1)$
Signed word	-32,768 to +32,767	-2^{15} to $(2^{15} - 1)$
Signed doubleword	-2,147,483,648 to 2,147,483,647	-2^{31} to $(2^{31} - 1)$
Signed quadword	-9,223,372,036,854,775,808 to +9,223,372,036,854,775,807	-2^{63} to $(2^{63} - 1)$

Representing Instructions

Representing Instructions	int sum(int x, int y)	Alpha sum	Sun sum	PC sum
	{	00	81	55
	return x+y;	00	C3	89
	}	30	E0	E5
- For this example, Alpha & Sun use two 4-byte instructions		42	08	8B
		01	90	45
		80	02	0C
		FA	00	03
		6B	09	45

08	89	EC	5D	C3
<i>Different machines use totally different instructions and encodings</i>				