Number Systems

Overview

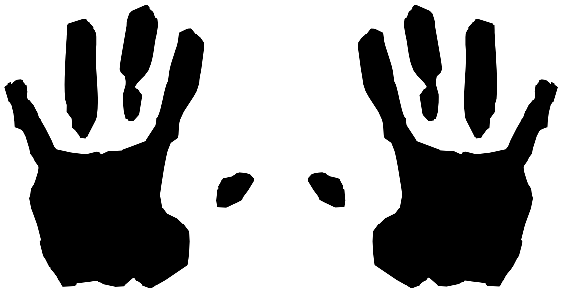
- The design of computers
  - It all starts with numbers
  - Building circuits
  - Building computing machines
- Digital systems
- Understanding decimal numbers
- Binary and octal numbers
  - The basis of computers!
- Conversion between different number systems

Digital Computer Systems

- **Digital systems consider *discrete* amounts of data.**
- **Examples**
  - **26 letters in the alphabet**
  - **10 decimal digits**
- **Larger quantities can be built from discrete values:**
  - **Words made of letters**
  - **Numbers made of decimal digits (e.g. 239875.32)**
- **Computers operate on *binary* values (0 and 1)**
- **Easy to represent binary values electrically**
  - **Voltages and currents.**
  - **Can be implemented using circuits**
  - **Create the building blocks of modern computers**

Understanding Decimal Numbers

- **Decimal numbers are made of decimal digits: (0,1,2,3,4,5,6,7,8,9)**
- **But how many items does a decimal number represent?**
  - **$8653 = 8x10^3 + 6x10^2 + 5x10^1 + 3x10^0$**
- **What about fractions?**
  - **$97654.35 = 9x10^4 + 7x10^3 + 6x10^2 + 5x10^1 + 4x10^0 + 3x10^{-1} + 5x10^{-2}$**
  - **In formal notation -> $(97654.35)_{10}$**
- **Why do we use 10 digits, anyway?**



Understanding Octal Numbers

- Octal numbers are made of octal digits: (0,1,2,3,4,5,6,7)
- How many items does an octal number represent?
  - $(4536)_8 = 4x8^3 + 5x8^2 + 3x8^1 + 6x8^0 = (1362)_{10}$
- What about fractions?
  - $(465.27)_8 = 4x8^2 + 6x8^1 + 5x8^0 + 2x8^{-1} + 7x8^{-2}$
- Octal numbers don't use digits 8 or 9
- Who would use octal number, anyway?

Understanding Binary Numbers

- Binary numbers are made of <u>b</u>inary dig<u>it</u>s (bits):
  - 0 and 1
- How many items does an binary number represent?
  - $(1011)_2 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = (11)_{10}$
- What about fractions?
  - $(110.10)_2 = 1x2^2 + 1x2^1 + 0x2^0 + 1x2^{-1} + 0x2^{-2}$
- Groups of eight bits are called a *byte*
  - $(11001001)_2$
- Groups of four bits are called a *nibble.*
  - $(1101)_2$

Why Use Binary Numbers?

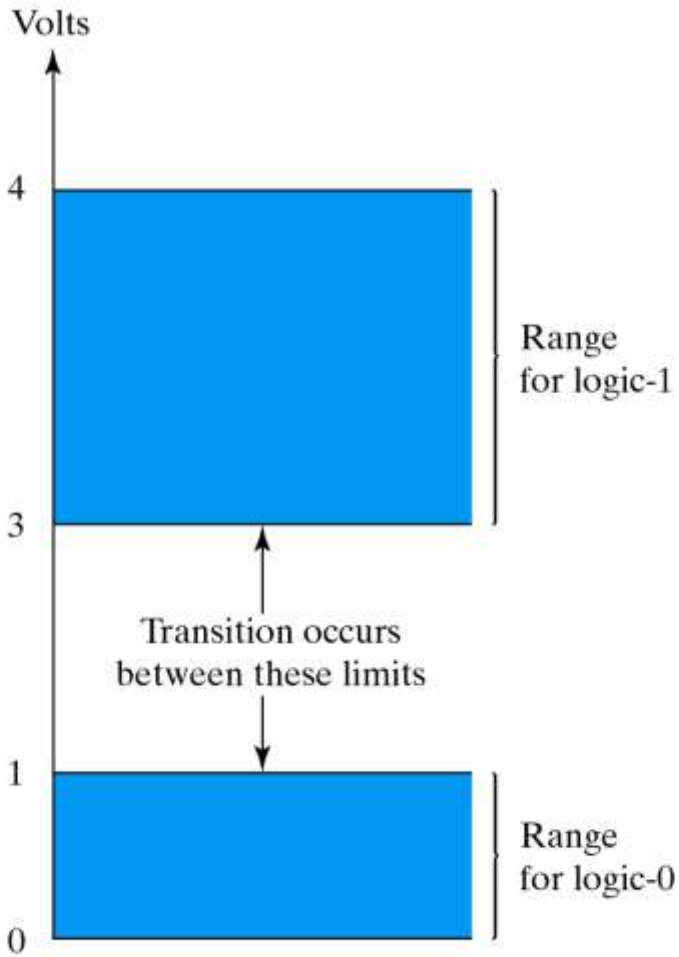
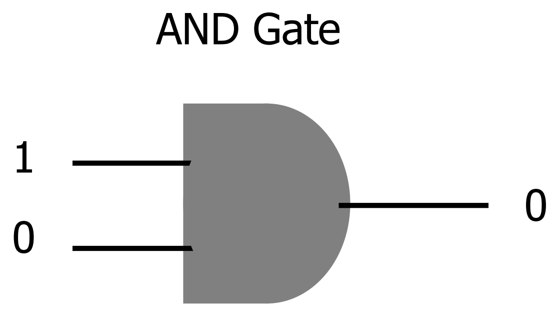
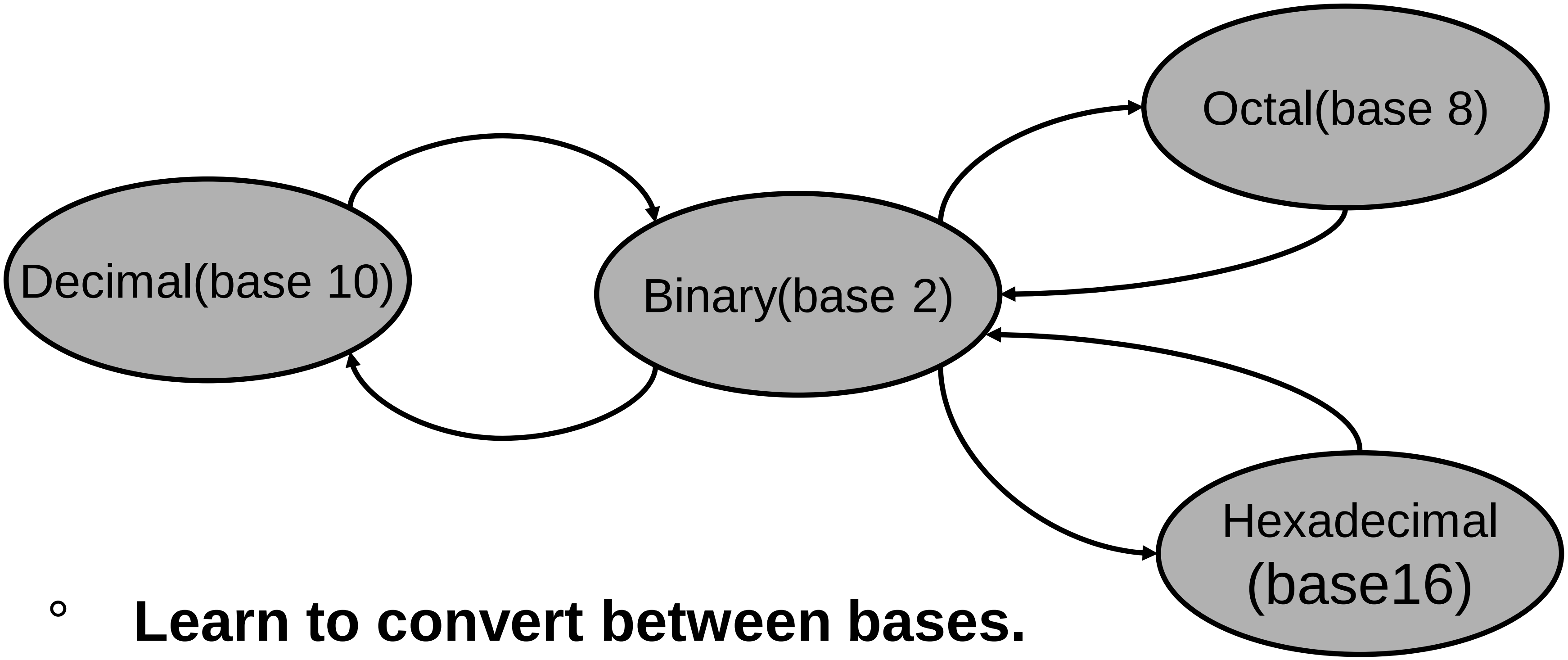


Fig. 1-3 Example of binary signals

- Easy to represent 0 and 1 using electrical values.
- Possible to tolerate noise.
- Easy to transmit data
- Easy to build binary circuits.



Conversion Between Number Bases



- **Learn to convert between bases.**
- **Already demonstrated how to convert from binary to decimal.**
- **Hexadecimal described in next lecture.**

Convert an Integer *from* Decimal *to* Another Base

For each digit position:

1. **Divide decimal number by the base (e.g. 2)**
2. **The *remainder* is the lowest-order digit**
3. **Repeat first two steps until no *divisor* remains.**

Example for $(13)_{10}$:

	Integer Quotient		Remainder	Coefficient
13/2 =	6	+	½	$a_0 = 1$
6/2 =	3	+	0	$a_1 = 0$
3/2 =	1	+	½	$a_2 = 1$
1/2 =	0	+	½	$a_3 = 1$

Answer $(13)_{10} = (a_3\, a_2\, a_1\, a_0)_2 = (1101)_2$

Convert an Fraction *from* Decimal *to* Another Base

For each digit position:

1. **Multiply decimal number by the base (e.g. 2)**
2. **The *integer* is the highest-order digit**
3. **Repeat first two steps until fraction becomes zero.**

Example for $(0.625)_{10}$:

	Integer		Fraction	Coefficient
0.625 x 2 =	1	+	0.25	$a_{-1} = 1$
0.250 x 2 =	0	+	0.50	$a_{-2} = 0$
0.500 x 2 =	1	+	0	$a_{-3} = 1$

Answer $(0.625)_{10} = (0.a_{-1}\, a_{-2}\, a_{-3})_2 = (0.101)_2$

The Growth of Binary Numbers

n	2^n
0	$2^0=1$
1	$2^1=2$
2	$2^2=4$
3	$2^3=8$
4	$2^4=16$
5	$2^5=32$
6	$2^6=64$
7	$2^7=128$

n	2^n	
8	$2^8=256$	
9	$2^9=512$	
10	$2^{10}=1024$	
11	$2^{11}=2048$	
12	$2^{12}=4096$	
20	$2^{20}=1M$	Mega
30	$2^{30}=1G$	Giga
40	$2^{40}=1T$	Tera

Binary Addition

- Binary addition is very simple.
- This is best shown in an example of adding two binary numbers…

```
   1  1  1  1   1  1 <---------------- carries
      1  1  1  1  0  1
+        1  0  1  1  1
---------------------
   1  0  1  0   1  0  0
```

Binary Subtraction

- We can also perform subtraction (with borrows in place of carries).
- Let's subtract $(10111)_2$ from $(1001101)_2$…

```
           1               10        <------  borrows
       0  10  10   0    0  10

       1   0   0   1    1   0   1
-              1   0    1   1   1
---------------------------------
           1   1   0    1   1   0
```

Binary Multiplication

- Binary multiplication is much the same as decimal multiplication, except that the multiplication operations are much simpler…

```
              1   0   1   1   1
X                 1   0   1   0
-------------------------------
              0   0   0   0   0
          1   0   1   1   1
      0   0   0   0   0
  1   0   1   1   1
-------------------------------
  1   1   1   0   0   1   1   0
```

Convert an Integer *from* Decimal *to* Octal

For each digit position:

1. **Divide decimal number by the base (8)**
2. **The *remainder* is the lowest-order digit**
3. **Repeat first two steps until no *divisor* remains.**

Example for $(175)_{10}$:

	Integer Quotient		Remainder	Coefficient
175/8 =	21	+	7/8	$a_0 = 7$
21/8 =	2	+	5/8	$a_1 = 5$
2/8 =	0	+	2/8	$a_2 = 2$

Answer $(175)_{10} = (a_2\, a_1\, a_0)_2 = (257)_8$

Convert an Fraction *from* Decimal *to* Octal

For each digit position:

1. **Multiply decimal number by the base (e.g. 8)**
2. **The *integer* is the highest-order digit**
3. **Repeat first two steps until fraction becomes zero.**

Example for $(0.3125)_{10}$:

	Integer		Fraction	Coefficient
0.3125 x 8 =	2	+	5	$a_{-1} = 2$
0.5000 x 8 =	4	+	0	$a_{-2} = 4$

Answer $(0.3125)_{10} = (0.24)_8$

Summary

- Binary numbers are made of binary digits (bits)
- Binary and octal number systems
- Conversion between number systems
- Addition, subtraction, and multiplication in binary