1.5 Number representation

Two Key Concepts

- Two key concepts that have made computers fast, powerful and ubiquitous:
  - **Binary Representation**
  - **Electronic Switches**
- **Why binary Representation ?**

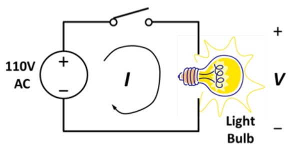
Decimal and Binary

- We are used to decimal representations, most likely because we have 10 fingers.
- However, since digital circuits respond to inputs and outputs that take on only the values {0, 1}, all arithmetic is performed using radix-2 or binary representation.

Circuit Models for Logic Function

<u>Switch</u>: Basic element of every digital (computer) system.

Circuit model:
Flip the lights
=> Voltage



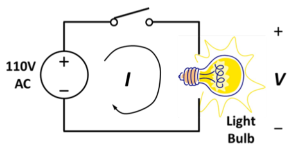
<u>Switch OPEN</u>: No closed path, V = 0
I = 0 => Light OFF.

<u>Switch CLOSED</u>: Closed path, Current flows
V = 110V, I = V/R => Light ON.

Circuit Models for Logic Function

<u>Switch</u>: Basic element of every digital (computer) system.

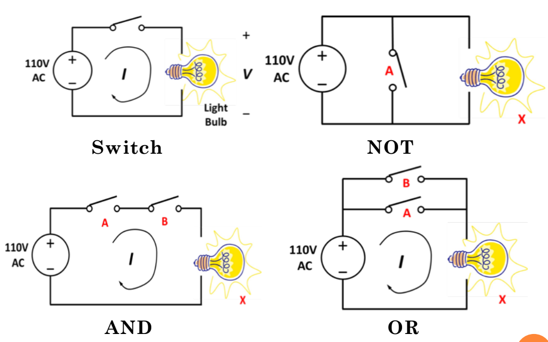
Circuit model:
Flip the lights
=> Voltage



Transistor: electrically controlled switch

- Turns ON/OFF by sensing voltage or current in circuit
- Allows logic, arithmetic to be implemented by digital hardware
- Advances in technology have made transistors smaller (more densely packed) and faster (more operations) every year.

Circuit Models for Logic Function



1.5 Binary Representation

- **All data in a computer is represented using 1's and 0's**
  - Numerical data: measurements, signals
  - Memory addresses
  - Instructions to CPU
  - Control signals: load register
  - Sound, images
  - Text

Example: $19 = 16 + 2 + 1 = 10011_2$

1.5 Binary Representation

- **All operations on data must be defined for binary operands.**
  - Arithmetic (Base 2, floating point)
  - Logical (AND, OR, NOT)
  - Signal + Image Processing

Example:

$$\begin{array}{rr} & \overset{1}{1}9 \\ + & 26 \\ \hline & 45 \end{array} \qquad \begin{array}{rr} & 1\,\overset{1}{0}\,0\,1\,1 \\ + & 1\,1\,0\,1\,0 \\ \hline & 1\,0\,1\,1\,0\,1 \end{array}$$

7

Digits and Radix

1) Numbers are usually written as a sequence of digits.

The radix or base associated with those digits tells us the value of each digit.

2) Let R be the Radix. For a number expressed with n digits

$d_{n-1}\, d_{n-2} \ldots d_2\, d_1\, d_0$, where $0 \le d_k \le R-1$,

the value of the number is

$$V_R = d_{n-1} R^{n-1} + d_{n-2} R^{n-2} + \ldots + d_2 R^2 + d_1 R^1 + d_0 R^0$$

8

Digits and Radix

Example: consider the digits **513**.

- For a decimal representation $R = 10$

  $513_{10} =$

- But for a base-7 representation

  $513_7 =$

9

Digits and Radix

Example: consider the digits **513**.

- For a decimal representation $R = 10$

  $513_{10} = (5 \cdot 100) + (1 \cdot 10) + (3 \cdot 1)$

- But for a base-7 representation

  $513_7 = (5 \cdot 49) + (1 \cdot 7) + (3 \cdot 1)$

  $= 245_{10} + 7_{10} + 3_{10}$

  $513_7 = 255_{10}$

10

Conversation between two different number systems

Binary → Decimal

Since we are familiar with decimal arithmetic, converting from binary to decimal is relatively easy.

Example: $\overset{8}{1}\overset{7}{0}\overset{6}{1}\overset{5}{1}\overset{4}{0}\overset{3}{0}\overset{2}{1}\overset{1}{0}\overset{0}{1}{}_2$

101100101_2

$= 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$

$= 256 + 64 + 32 + 4 + 1$

$= 357$

11

Conversation between two different number systems

Decimal → Binary

$$V_R = d_{n-1} R^{n-1} + d_{n-2} R^{n-2} + \ldots + d_2 R^2 + d_1 R^1 + d_0 R^0$$

Suppose we divide a number V by a radix R.

$\frac{V}{R}$: Quotient $= d_{n-1} R^{n-2} + d_{n-2} R^{n-3} + \ldots + d_2 R + d_1$

Remainder $= d_0$

Repeat:

Quotient $= d_{n-1} R^{n-3} + \ldots + d_2$

Remainder $= d_1$

- By repeatedly dividing the quotients by R, the sequence of remainders are the digits in a radix-R representation.

12

DECIMAL → BINARY

Converting from decimal to binary takes longer, because we still carry out the operations using decimal arithmetic.

- **Two methods**
  - Least Significant Bit (LSB) First
  - Most Significant Bit (MSB) First

LEAST SIGNIFICANT BIT (LSB) FIRST

$b_{n-1}\ b_{n-2}\ \ldots\ b_2\ b_1\ b_0$

LSB

- Step 1 – Divide the number by 2. Let q be the quotient, the remainder is b_0.
- Step 2 – Divide q by 2. The new remainder is the next bit (b_1). Replace q with the new quotient.
- Step 3 – Repeat Step 2 until quotient = 0.

LEAST SIGNIFICANT BIT (LSB) FIRST

Example: $948_{10} = (\overset{9876543210}{1110110100})_2$

	Quotient	Remainder	
948/2	474	0	← b_0 LSB
474/2	237	0	← b_1
237/2	118	1	← b_2
118/2	59	0	
59/2	29	1	
29/2	14	1	
14/2	7	0	
7/2	3	1	
3/2	1	1	
1/2	0	1	← b_9 MSB

MOST SIGNIFICANT BIT (MSB) FIRST

$b_{n-1}\ b_{n-2}\ \ldots\ b_2\ b_1\ b_0 = D_{10}$

MSB

- Step 1 – Find the largest integer n such that $2^n \le D$, then

  $b_n = 1$, and let $X = D - 2^n$
- Step 2 – Find the largest integer k such that $2^k \le X$ then

  $b_k = 1$, and replace X with $X - 2^k$.
- Step 3 – Repeat Step 2 until $X = 0$.

MOST SIGNIFICANT BIT (MSB) FIRST

Example: 948_{10}

$2^9 = 512$, $2^{10} = 1024$

	→	$b_9 = 1$;	$X = 948 - 512 = 436$
$2^8 = 256$	→	$b_8 = 1$;	$X = 436 - 256 = 180$
$2^7 = 128$	→	$b_7 = 1$;	$X = 180 - 128 = 52$
$2^6 = 64$	→	$b_6 = 0$;	
$2^5 = 32$	→	$b_5 = 1$;	$X = 52 - 32 = 20$
$2^4 = 16$	→	$b_4 = 1$;	$X = 20 - 16 = 4$
$2^3 = 8$	→	$b_3 = 0$;	
$2^2 = 4$	→	$b_2 = 1$;	$X = 4 - 4 = 0$
	→	$b_1 = 0$;	
	→	$b_0 = 0$.	

QUIZ 1:

- **Convert the radix-13 number 133_{13} to the equivalent decimal and binary representations.**

- Assignment: Page 18-19: 1.2, 1.3, 1.6

Octal and Hexadecimal

- In computers, data, memory addresses, operation codes, etc. may be represented using 16, 32, 64, or more bits. It is awkward for people to deal with such long bit strings.
- Therefore, binary data are often reported in
  - **Base 8 (Octal), or**
  - **Base 16 (Hexadecimal)**.
- Conversion between binary, octal and hexadecimal is easy.

Octal and Hexadecimal

Suppose we have a 12-bit number

$$b_{11}\, b_{10}\, b_9\, b_8\, b_7\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0$$

$$\begin{aligned} V = \; & b_{11}\cdot 2^{11} + b_{10}\cdot 2^{10} + b_9\cdot 2^9 + b_8\cdot 2^8 + b_7\cdot 2^7 \\ & + b_6\cdot 2^6 + b_5\cdot 2^5 + b_4\cdot 2^4 + b_3\cdot 2^3 + b_2\cdot 2^2 + b_1\cdot 2 + b_0 \\ = \; & \underbrace{(b_{11}\cdot 2^2 + b_{10}\cdot 2 + b_9)}_{O_3}\cdot 2^9 + \underbrace{(b_8\cdot 2^2 + b_7\cdot 2 + b_6)}_{O_2}\cdot 2^6 \\ & + \underbrace{(b_5\cdot 2^2 + b_4\cdot 2 + b_3)}_{O_1}\cdot 2^3 + \underbrace{(b_2\cdot 2^2 + b_1\cdot 2 + b_0)}_{O_0} \end{aligned}$$

$\rightarrow$ $\boldsymbol{V = O_3\cdot 8^3 + O_2\cdot 8^2 + O_1\cdot 8 + O_0}$

- Each octal digit is formed from 3 bits.

Octal and Hexadecimal

Similarly,

$$\begin{aligned} V = \; & b_{11}\cdot 2^{11} + b_{10}\cdot 2^{10} + b_9\cdot 2^9 + b_8\cdot 2^8 + b_7\cdot 2^7 \\ & + b_6\cdot 2^6 + b_5\cdot 2^5 + b_4\cdot 2^4 + b_3\cdot 2^3 + b_2\cdot 2^2 + b_1\cdot 2 + b_0 \\ = \; & \underbrace{(b_{11}\cdot 2^3 + b_{10}\cdot 2^2 + b_9\cdot 2 + b_8)}_{h_2}\cdot 2^8 \\ & + \underbrace{(b_7\cdot 2^3 + b_6\cdot 2^2 + b_5\cdot 2 + b_4)}_{h_1}\cdot 2^4 \\ & + \underbrace{(b_3\cdot 2^3 + b_2\cdot 2^2 + b_1\cdot 2 + b_0)}_{h_0}\cdot 2^0 \end{aligned}$$

$\rightarrow$ $\boldsymbol{V = h_2\cdot 16^2 + h_1\cdot 16 + h_0}$

- Each hexadecimal digit is formed from 4 bits.

Octal and Hexadecimal

Example

$526_8 = 5 \cdot 8^2 + 2 \cdot 8 + 6 \cdot 1$

$5_8 = 4\cdot 1 + 2\cdot 0 + 1\cdot 1 = 101_2$

$2_8 = 4\cdot 0 + 2\cdot 1 + 0\cdot 1 = 010_2$

$6_8 = 4\cdot 1 + 2\cdot 1 + 1\cdot 0 = 110_2$

$\rightarrow$ $\boldsymbol{526_8 = 101 \mid 010 \mid 110_2 = 101010110_2}$

Divide into 4-bit groups 1 | 0101 | 0110

$0110_2 = 0\cdot 8 + 1\cdot 4 + 1\cdot 2 + 0\cdot 1 = 6_{16}$

$0101_2 = 0\cdot 8 + 1\cdot 4 + 0\cdot 2 + 1\cdot 1 = 5_{16}$

$0001_2 = 0\cdot 8 + 0\cdot 4 + 0\cdot 2 + 1\cdot 1 = 1_{16}$

$\rightarrow$ $\boldsymbol{526_8 = 156_{16}}$

Hexadecimal

- The letters A – F are used to represent 10 – 15.

Dec.	Bin.	Hex.
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

Dec.	Bin.	Hex.
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Octal and Hexadecimal

Example:

$F2A_{16} = 15_{10}\cdot 16^2 + 2_{10}\cdot 16 + 10_{10}$

$= 3840 + 32 + 10$

$F2A_{16} = 3882_{10}$

$F_{16} = 1111_2$

$2_{16} = 0010_2$

$A_{16} = 1010_2$

$010_2 = 2$

$101_2 = 5$

$100_2 = 4$

$111_2 = 7$

$\boldsymbol{F2A_{16}} = 1111 \mid 0010 \mid 1010_{\,2}$

$= 111 \mid 100 \mid 101 \mid 010_2$ $\boldsymbol{= 7452_8}$

Decimal → Octal or Hexadecimal

- To convert from decimal to octal or hexadecimal, it is usually easiest to cover to binary first.

Example: 737_{10} = 1011100001_2

	Quotient	Remainder	
737/2	368	1	b_0
368/2	184	0	b_1
184/2	92	0	b_2
92/2	46	0	b_3
46/2	23	0	b_4
23/2	11	1	b_5
11/2	5	1	b_6
5/2	2	1	b_7
2/2	1	0	b_8
1/2	0	1	b_9

25

Decimal → Octal or Hexadecimal

- To convert from decimal to octal or hexadecimal, it is usually easiest to cover to binary first.

Example: 737_{10} = 1011100001_2

Octal:

1 | 011 | 100 | 001

↓ ↓ ↓ ↓

1 3 4 1

$737_{10} = 1341_8$

Hex:

10 | 1110 | 0001

↓ ↓ ↓

2 14=E 1

$737_{10} = 2E1_{16}$

26

Decimal → Octal or Hexadecimal

- **Direct Conversion**

737_{10} → Octal

	Quotient	Remainder
737/8	92	1
92/8	11	4
11/8	1	3
1/8	0	1

$737_{10} = 1341_8$

737_{10} → Hexadecimal

	Quotient	Remainder
737/16	46	1
46/16	2	E (14_{10})
2/16	0	2

$737_{10} = 2E1_{16}$

27

Octal or Hexadecimal → Decimal

- Direct Conversion

Example: 10422_5 → Decimal

$$\begin{aligned} 10422_5 &= 1\cdot 5^4 + 4\cdot 5^2 + 2\cdot 5 + 2\cdot 1 \\ &= 625 + 100 + 10 + 2 \\ &= 737_{10} \end{aligned}$$

28

ASICII Character Code

Table 1.2 The seven-bit ASCII code.

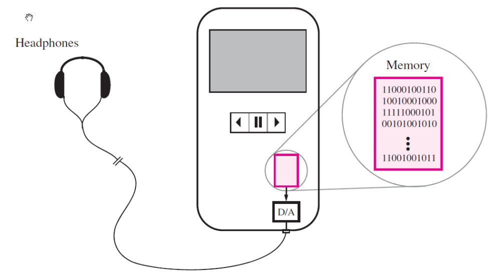
Bit positions 3210	Bit positions 654: 000	001	010	011	100	101	110	111
0000	NUL	DLE	SPACE	0	@	P	´	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	—	o	DEL

NUL	Null/Idle	SI	Shift in
SOH	Start of header	DLE	Data link escape
STX	Start of text	DC1-DC4	Device control
ETX	End of text	NAK	Negative acknowledgement
EOT	End of transmission	SYN	Synchronous idle
ENQ	Enquiry	ETB	End of transmitted block
ACQ	Acknowledgement	CAN	Cancel (error in data)
BEL	Audible signal	EM	End of medium
BS	Back space	SUB	Special sequence
HT	Horizontal tab	ESC	Escape
LF	Line feed	FS	File separator
VT	Vertical tab	GS	Group separator
FF	Form feed	RS	Record separator
CR	Carriage return	US	Unit separator
SO	Shift out	DEL	Delete/Idle

Bit positions of code format = 6 5 4 3 2 1 0

29

Digital and Analog Information



30

Digital and Analog Information

Analog and **digital** signals are used to transmit information, usually through electric signals.

In both these technologies, the information, such as any audio or video, is transformed into electric signals.

The **difference between analog and digital** technologies is that in analog technology, information is translated into electric pulses of varying amplitude. In digital technology, translation of information is into binary format (zero or one) where each bit is representative of two distinct amplitudes.

31

Conclusion

- Digital Representation of information
  - Binary to decimal
  - Decimal to binary
- Textbook Reading: 1.1;1.2;1.3;1.4;1.5; 3.1.2
- Assignment: Page 18-19: 1.2, 1.3, 1.6

32

END

33