

Lógica Digital (1001351)

Representação Digital da Informação

Prof. Edilson Kato
kato@ufscar.br

Prof. Maurício Figueiredo
mauricio@ufscar.br

Prof. Ricardo Menotti
menotti@ufscar.br

Prof. Roberto Inoue
rsinoue@ufscar.br

Departamento de Computação
Universidade Federal de São Carlos

Atualizado em: 27 de fevereiro de 2019



Representação Digital da Informação

- ▶ Nos circuitos lógicos a informação é representada como sinais eletrônicos;
- ▶ Pode-se considerar que cada sinal representa um dígito de informação;
- ▶ Para tornar o projeto de circuitos lógicos mais fácil e preciso, cada dígito pode assumir apenas dois estados:
 - ▶ 0 (zero) ou 1 (um);
 - ▶ L (low) ou H (high);
 - ▶ F (false) ou T (true);

Representação Digital da Informação

- ▶ Esses valores lógicos são implementados como níveis de tensão em um circuito:
 - ▶ o valor 0 é geralmente representado como 0 V (terra);
 - ▶ o valor 1 é a tensão nível da fonte de alimentação do circuito (normalmente entre 1 e 5 V CC).
- ▶ Em geral, todas as informações nos circuitos lógicos são representadas como combinações de 0s e 1s;
- ▶ Antes de iniciar a discussão de circuitos lógicos, será útil examinar como números, dados alfanuméricos (texto) e outras informações podem ser representados usando os dígitos 0 e 1.

Números Decimais

- ▶ No sistema decimal, um número consiste em dígitos que têm 10 valores possíveis, de 0 a 9, e cada dígito representa um múltiplo de uma potência de 10.
 - ▶ Por exemplo, o número 8547 representa
$$8 \times 10^3 + 5 \times 10^2 + 4 \times 10^1 + 7 \times 10^0$$
 - ▶ Normalmente não escrevemos as potências de 10, pois estão implícitas nas posições.
 - ▶ Isso é chamado de representação numérica posicional.
- ▶ Formalmente:
 - ▶ $D = d_{n-1}d_{n-2}\dots d_1d_0$
 - ▶ $V(D) = d_{n-1} \times 10^{n-1} + d_{n-2} \times 10^{n-2} \dots d_1 \times 10^1 + d_0 \times 10^0$
- ▶ Como os dígitos têm 10 valores possíveis e cada dígito é podendera como uma potência de 10, dizemos que os números decimais são números de base-10.

Números Binários

- ▶ Como os circuitos digitais representam informações usando somente os valores 0 e 1, não é prático ter dígitos que possam assumir dez valores;
- ▶ Nesses circuitos, é mais apropriado usar o sistema binário, ou base-2, que possui apenas os dígitos 0 e 1.
- ▶ Cada dígito binário é chamado de bit.
- ▶ No sistema numérico binário, a mesma representação numérica posicional é usada:

- ▶ $B = b_{n-1}b_{n-2}...b_1b_0$

- ▶ $V(B) = b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} ... b_1 \times 2^1 + b_0 \times 2^0$

- ▶ $= \sum_{i=0}^{n-1} b_i \times 2^i$

Números Binários

- ▶ Por exemplo, o número binário 1101 representa o valor
 - ▶ $V = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
- ▶ Como um determinado padrão de dígitos tem significados diferentes para bases diferentes, indicaremos a base como um subscrito quando houver potencial para confusão.
- ▶ Assim, para especificar que 1101 é um número de base 2, escreveremos $(1101)_2$.
- ▶ Avaliar a expressão anterior para V fornece:
 - ▶ $V = 8 + 4 + 0 + 1 = 13$
- ▶ Consequentemente:
 - ▶ $(1101)_2 = (13)_{10}$

Números Binários

- O intervalo de inteiros que pode ser representado por um número binário depende do número de bits usados:

Representação Decimal	Representação Binária
00	0000
01	0001
02	0010
03	0011
04	0100
05	0101
06	0110
07	0111
08	1000
09	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Números Binários

- ▶ Um exemplo de um número maior é $(10110111)_2 = (183)_{10}$;
- ▶ Em geral, o uso de n bits permite a representação de inteiros positivos no intervalo de 0 a $2^n - 1$;
- ▶ Em um número binário, o bit mais à direita é geralmente referido como o **bit menos significativo** (LSB¹);
- ▶ O bit mais à esquerda, que possui a maior potência de 2 associada a ele, é chamado de **bit mais significativo** (MSB²).
- ▶ Em sistemas digitais, muitas vezes é conveniente considerar vários bits juntos como um grupo:
 - ▶ Um grupo de **quatro bits** é chamado de *nibble*;
 - ▶ Um grupo de **oito bits** é chamado de *byte*.

¹ *least significant bit*

² *most significant bit*

Conversão entre os sistemas decimal e binário

- ▶ Um número binário é convertido em um número decimal simplesmente aplicando a equação a seguir:
 - ▶ $V = b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} \dots b_1 \times 2^1 + b_0 \times 2^0$
- ▶ Converter um número decimal em um número binário não é tão simples, porque precisamos construir o número usando potências de 2.
- ▶ Por exemplo, o número $(17)_{10}$ é $2^4 + 2^0 = (10001)_2$ e o número $(50)_{10}$ é $2^5 + 2^4 + 2^1 = (110010)_2$. Em geral, a conversão pode ser realizada dividindo sucessivamente o número decimal por 2.

Conversão entre os sistemas decimal e binário

► Converter $(857)_{10}$:

		Resto	
$857 \div 2 =$	428	1	LSB
$428 \div 2 =$	214	0	
$214 \div 2 =$	107	0	
$107 \div 2 =$	53	1	
$53 \div 2 =$	26	1	
$26 \div 2 =$	13	0	
$13 \div 2 =$	6	1	
$6 \div 2 =$	3	0	
$3 \div 2 =$	1	1	
$1 \div 2 =$	0	1	MSB

► Resultado $(1101011001)_2$

- Note que o LSB é gerado primeiro e o MSB é gerado por último;
- Estamos considerando apenas a representação de inteiros positivos, depois veremos outras e suas operações aritméticas.

Tabela ASCII de caracteres

- ▶ As informações alfanuméricas, como letras e números digitados em um teclado de computador, são representadas como códigos de 0 e 1 dígitos;
- ▶ O código mais comum usado para esse tipo de informação é conhecido como o código ASCII³.

³ *American Standard Code for Information Interchange*

Tabela ASCII de caracteres

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

Tabela ASCII de caracteres

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
---	---	---	---	---	---	---

Tabela ASCII de caracteres

- ▶ O código ASCII usa padrões de sete bits para denotar 128 caracteres diferentes;
- ▶ Dez dos caracteres são dígitos decimais de 0 a 9, os bits de alta ordem têm o mesmo padrão, $b_6b_5b_4 = 011$, para todos os 10 dígitos e cada dígito é identificado pelos quatro bits de ordem baixa, b_{3-0} , usando os padrões binários para esses dígitos;
- ▶ Letras maiúsculas e minúsculas são codificadas de uma maneira que facilita a classificação de informações textuais. Os códigos de A a Z estão em sequência numérica ascendente, o que significa que a tarefa de ordenar letras (ou palavras) pode ser realizada por uma simples comparação aritmética dos códigos que representam as letras;

Tabela ASCII de caracteres

- ▶ Além de códigos que representam caracteres e letras, o código ASCII inclui sinais de pontuação como ! e ?, símbolos comumente usados, tais como & e % e uma coleção de caracteres de controle;
- ▶ O padrão ASCII usa sete bits para codificar um caractere.

Informação Digital e Analógica

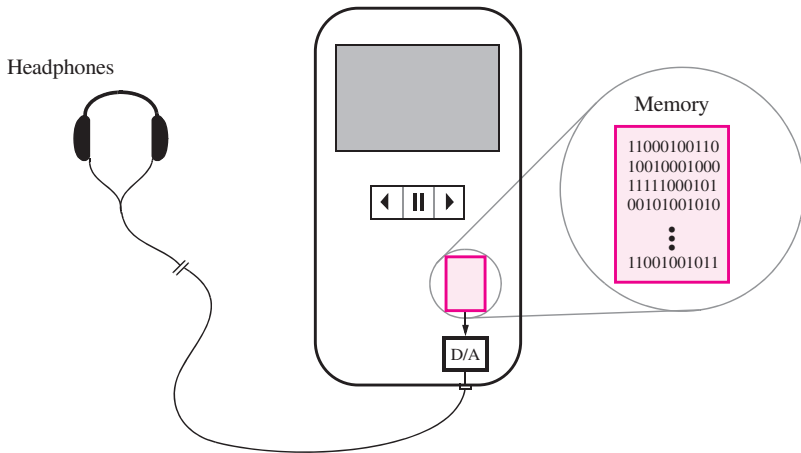


Figure 1.7 Using digital technology to represent music.

Teoria e Prática

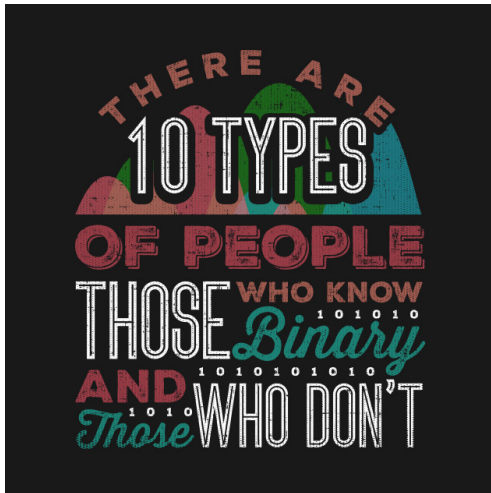
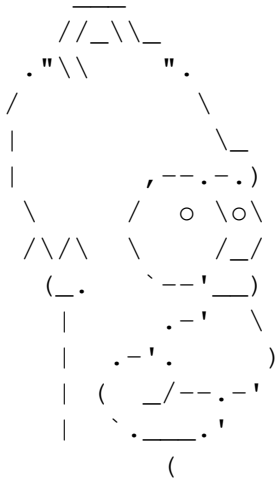
- ▶ Existem numerosas técnicas para lidar com circuitos lógicos;
- ▶ A álgebra booleana, que veremos mais adiante, foi adotada como um meio matemático para representar tais circuitos;
- ▶ As ferramentas de CAD⁴ não só tornaram possível projetar circuitos incrivelmente complexos, mas também tornaram o projeto muito mais simples, pois realizam muitas tarefas automaticamente;

⁴ *computer aided design*

Teoria e Prática

- ▶ Por que não aprender simplesmente como usar as ferramentas CAD?
 - ▶ Elas partem de descrições que, se forem mal especificadas, gerem circuitos de baixa qualidade;
 - ▶ Não é possível compreender o que as ferramentas fazem sem conhecer a teoria subjacente;
 - ▶ Elas oferecem muitas etapas opcionais que devem ou não serem usadas em uma determinada situação;
- ▶ **Não é possível tornar-se um projetista de circuitos lógicos efetivo sem entender os conceitos fundamentais!**

D'oh!



Bibliografia

- ▶ Brown, S. & Vranesic, Z. - Fundamentals of Digital Logic with Verilog Design, 3rd Ed., Mc Graw Hill, 2009
- ▶ <https://www.asciart.eu/>

Lógica Digital (1001351)

Representação Digital da Informação

Prof. Edilson Kato
kato@ufscar.br

Prof. Maurício Figueiredo
mauricio@ufscar.br

Prof. Ricardo Menotti
menotti@ufscar.br

Prof. Roberto Inoue
rsinoue@ufscar.br

Departamento de Computação
Universidade Federal de São Carlos

Atualizado em: 27 de fevereiro de 2019

