

Lógica de programação

Aula introdutória (parte 2)

Walisson Pereira

walisson_pereira@uvanet.br

Universidade Estadual Vale do Acaraú

Estrutura básica de um computador

Representação dos dados

Referências

Estrutura básica de um computador

Um computador possui duas partes diferentes que trabalham juntas:

- **hardware**: composto pelas partes físicas
- **software**: composto pelos programas

Estrutura básica de um computador

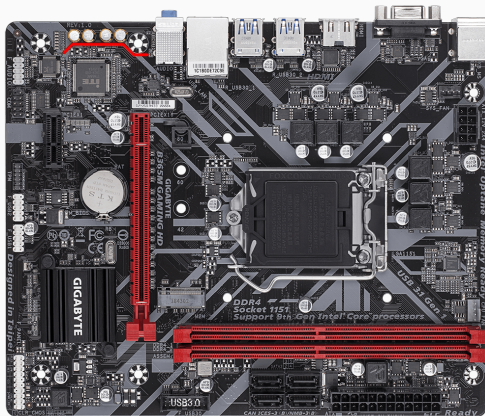


Figura 1: Placa-mãe gigabyte B365 [1]

Estrutura básica de um computador

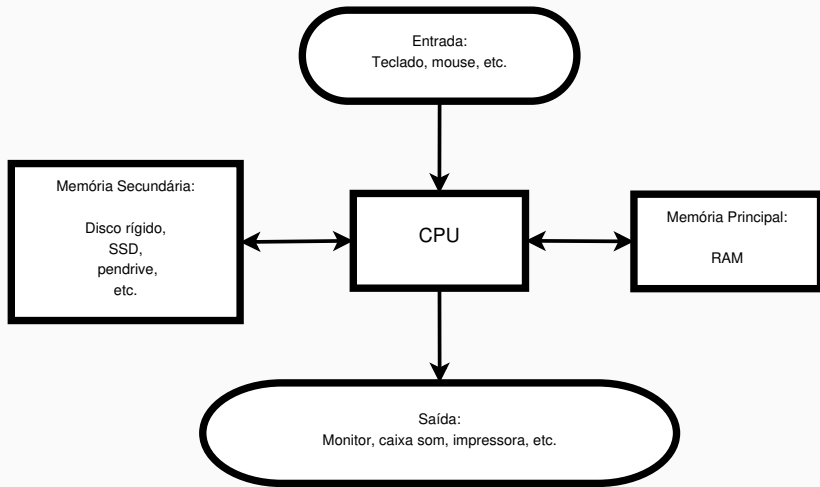


Figura 2: Esquema simplificado de um computador

Estrutura básica de um computador

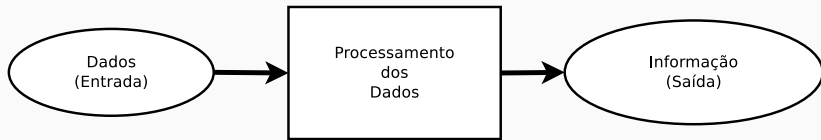


Figura 3: Fluxo de dados

Os computadores são capazes de manipular e armazenar um grande volume de dados que vão resultar as informações que precisamos.

Representação dos dados

- A informação é fornecida a partir da presença ou ausência de uma tensão elétrica.
- Caracterizamos estes dois estados de 0 e 1 (binário).

Representação dos dados

A informação é manipulada através da combinação de transistores que são o elemento primário para a construção dos circuitos integrados de um computador.



Figura 4: Exemplo de um transistor [2]

Representação dos dados

- Uma CPU atual possui em torno de 3 bilhões de transístores.



Figura 5: A CPU Intel 8086 e a CPU Intel Core i7 8086K [3]

Representação dos dados

Uma CPU Intel 8086 possui 879 transístores por milímetro quadrado [3].

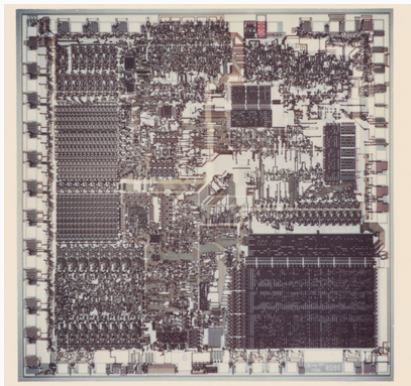


Figura 6: O die de uma CPU Intel 8086 [3]

Representação dos dados

Uma CPU Intel i7 8086K possui em torno de 20 milhões de transístores por milímetro quadrado [3].

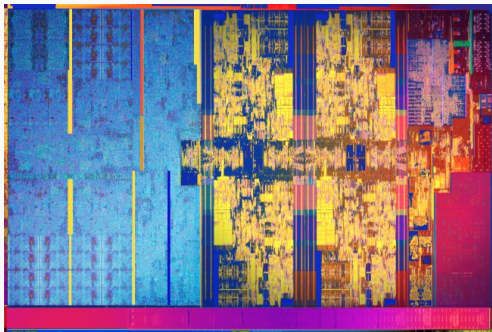


Figura 7: O die de uma CPU Intel Coffe Lake [3]

Voltando ao assunto:

- É necessário que as informações estejam em um formato binário.

Lei de formação dos números

- $317 = 3 \times 100 + 1 \times 10 + 7 \times 1$

Lei de formação dos números

- $317 = 3 \times 100 + 1 \times 10 + 7 \times 1$
- $317 = 3 \times 10^2 + 1 \times 10^1 + 7 \times 10^0$

Convertendo um binário em número inteiro

- $(100101)_2$

Convertendo um binário em número inteiro

- $(100101)_2$
- $(1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

Convertendo um binário em número inteiro

- $(100101)_2$
- $(1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$
- $(1 \times 32) + (0 \times 16) + (0 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1)$

Convertendo um binário em número inteiro

- $(100101)_2$
- $(1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$
- $(1 \times 32) + (0 \times 16) + (0 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1)$
- $32 + 0 + 0 + 4 + 0 + 1$

Convertendo um binário em número inteiro

- $(100101)_2$
- $(1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$
- $(1 \times 32) + (0 \times 16) + (0 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1)$
- $32 + 0 + 0 + 4 + 0 + 1$
- 37

Tabela para conversão

32	16	8	4	2	1
1	0	0	1	0	1

Convertendo um número inteiro para binário

37

$37 \div 2 \rightarrow 18$ resta 1

$18 \div 2 \rightarrow 9$ resta 0

$9 \div 2 \rightarrow 4$ resta 1

$4 \div 2 \rightarrow 2$ resta 0

$2 \div 2 \rightarrow 1$ resta 0

$1 \div 2 \rightarrow 0$ resta 1

Resultando $(100101)_2$

- Representamos o número 37 com o número binário $(100101)_2$
- Chamamos cada dígito binário de **bit**
- Logo, para representar o número 37, precisamos de 6 bits.

Quantas informações consigo representar com 6 bits?

- $2^6 = 64$ combinações diferentes.

Representação dos dados

Podemos também representar outras informações:

000 (nul)	016 ► (dle)	032 sp	048 0	064 @	080 P	096 `	112 p
001 ☉ (soh)	017 ◀ (dc1)	033 !	049 1	065 A	081 Q	097 a	113 q
002 ● (stx)	018 † (dc2)	034 "	050 2	066 B	082 R	098 b	114 r
003 ▼ (etx)	019 !! (dc3)	035 #	051 3	067 C	083 S	099 c	115 s
004 ♦ (eot)	020 ¶ (dc4)	036 \$	052 4	068 D	084 T	100 d	116 t
005 ♣ (enq)	021 § (nak)	037 %	053 5	069 E	085 U	101 e	117 u
006 ♠ (ack)	022 — (syn)	038 &	054 6	070 F	086 V	102 f	118 v
007 • (bel)	023 ‡ (etb)	039 '	055 7	071 G	087 W	103 g	119 w
008 ▣ (bs)	024 † (can)	040 (056 8	072 H	088 X	104 h	120 x
009 (tab)	025 ↓ (em)	041)	057 9	073 I	089 Y	105 i	121 y
010 (lf)	026 (eof)	042 *	058 :	074 J	090 Z	106 j	122 z
011 ♂ (vt)	027 ← (esc)	043 +	059 ;	075 K	091 [107 k	123 {
012 ♀ (np)	028 L (fs)	044 ,	060 <	076 L	092 \	108 l	124
013 (cr)	029 ↔ (gs)	045 -	061 =	077 M	093]	109 m	125 }
014 ♀ (so)	030 ▲ (rs)	046 .	062 >	078 N	094 ^	110 n	126 ~
015 ☆ (si)	031 ▼ (us)	047 /	063 ?	079 O	095 _	111 o	127 ∆

Figura 8: Tabela ASCII

8 bits (b) = 1 Byte (B)

Quantas informações eu consigo guardar em:

- 8 bits $\rightarrow 2^8 = 256$
- 16 bits $\rightarrow 2^{16} = 65536$
- 32 bits $\rightarrow 2^{32} = 4294967296$
- 64 bits $\rightarrow 2^{64} = 1,844674407 \times 10^{19}$

Referências

- 1 Placa-mãe gigabyte B365M H (rev. 1.0) (<https://www.gigabyte.com/br/Motherboard/B365M-H-rev-10#kf>. Acesso em 04 de setembro de 2020.
- 2 Transistor NPN - BC546. <https://www.baudaeletronica.com.br/transistor-npn-bc546.html>. Acesso em 04 de setembro de 2020.
- 3 ALCORN, P. The Core i7-8086k Review: 40 years of x86. 2018. <https://www.tomshardware.com/reviews/intel-core-i7-8086k-cpu-8086-anniversary,5658-2.html>. Acesso em 04 de setembro de 2020.