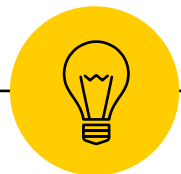
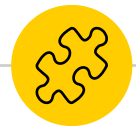


Arquitetura de Computadores (AC22S)

A Informação e sua Representação



Gustavo Santos



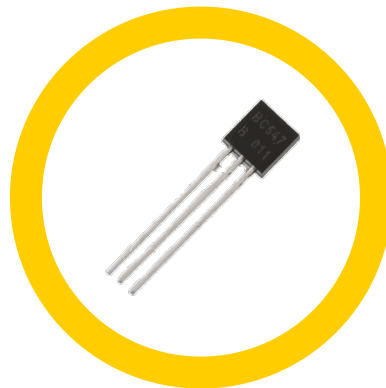
Recapitulação — Gerações



1947—1957

Primeira Geração
Válvulas

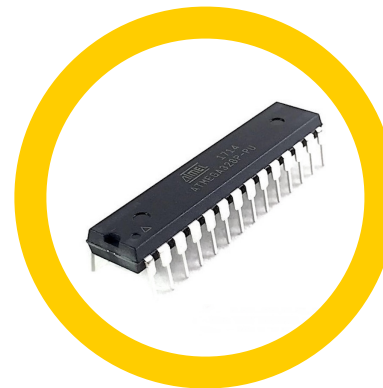
Mark I, ENIAC



1956—1965

Segunda Geração
Transistores

UNIVAC



1964—1975

Terceira Geração
Circuitos integrados

IBM 360

Quarta Geração
Integração em escala
muito grande

Quinta Geração
Computadores de
baixa potência



Representação

O computador, sendo um equipamento eletrônico, armazena e movimenta as informações a partir de dois estados distintos: ligado e desligado. Seja via eletricidade, polaridade magnética ou pela luz refletida.

Nesse sentido, computadores digitais trabalham com dois sinais elétricos: alto e baixo, 0 ou 1.

Como representar texto, imagens, números?!



Representações diversas

Informação	#Representações
Caracteres alfabéticos	26
Algarismos decimais	10
Sinais de Pontuação	32
Caracteres de controle	24
Total	118



Representação binária

- A conversão de dados (0 e 1) em informações (e vice-versa) é fundamental na compreensão de como um computador funciona
- Endereços de memória, instruções, etc. são representados internamente como 0s e 1s



Notação Posicional

Nos sistemas numéricos como o decimal, o valor de um dígito depende da posição que o dígito ocupa em um número. Exemplo:

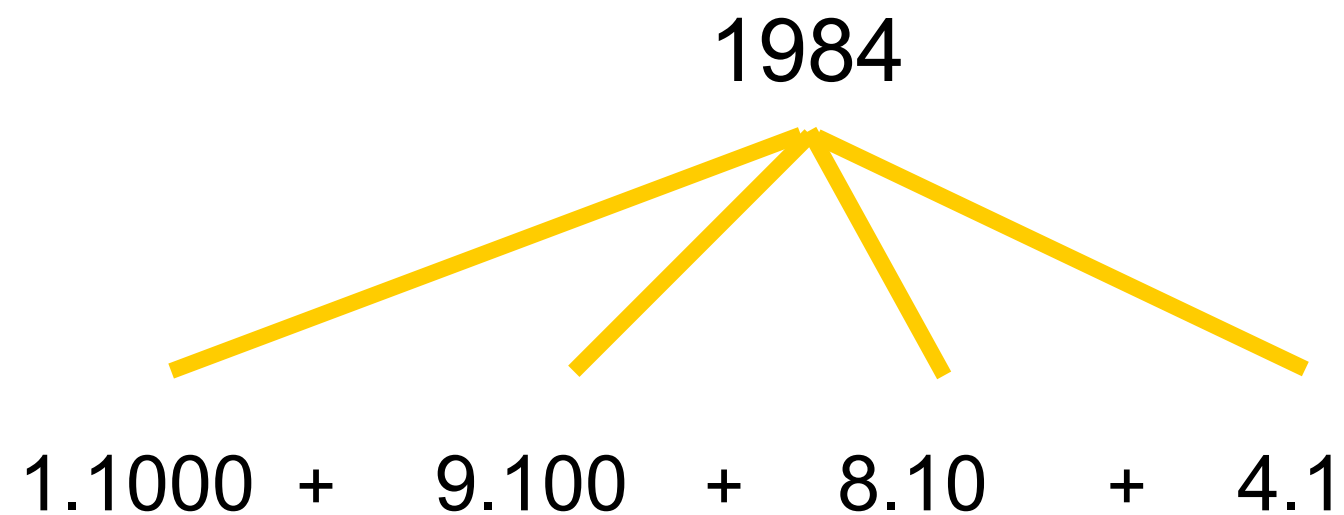
1984

Dessa forma, “lemos” um número da esquerda para a direita, do dígito com maior valor para o menor.



Notação Posicional

- O valor total do número então é a soma dos valores relativos a cada posição





Notação Posicional

- O valor total do número então é a soma dos valores relativos a cada posição
- Não por acaso, podemos decompor este número em potências da base utilizada (no caso, 10)

$$1984 = 1 \cdot 10^3 + 9 \cdot 10^2 + 8 \cdot 10^1 + 4 \cdot 10^0$$



Lei de Formatação

- De uma forma mais geral, podemos decompor um número de qualquer base numérica dessa forma

$$a_n * b^n + a_{n-1} * b^{n-1} + \dots + a_1 * b^1 + a_0 * b^0$$

- Onde a é o algarismo, b a base numérica, e n a posição do algarismo; da posição com mais valor, para a posição com menos valor



Exercício de fixação

Representar o número $(84917)_{10}$ usando a lei de formatação



Exercício de fixação

base 10

Representar o número $(84917)_{10}$ usando a lei de formatação

$$\begin{aligned} &8*10^4 + 4*10^3 + 9*10^2 + 1*10^1 + 7*10^0 \\ &80000 + 4000 + 900 + 10 + 7 \\ &(84917)_{10} \end{aligned}$$



Sistema de Numeração

Conjunto de símbolos utilizados para a representação de informação e regras que definem a forma de representação

A informação não muda, apenas os símbolos utilizados para representar aquela informação

Como representar $(10110)_2$ num sistema decimal?



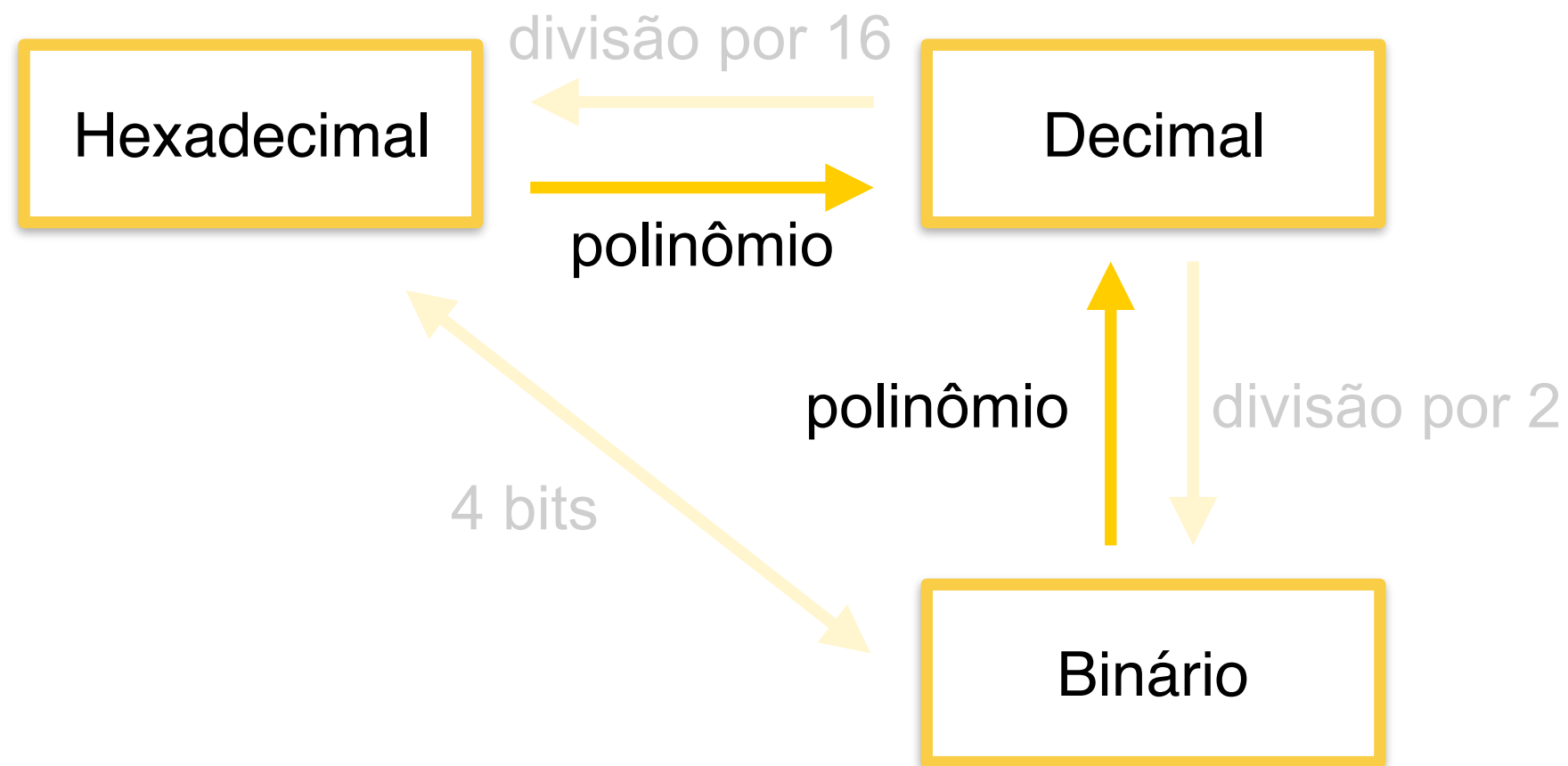
Sistemas de Numeração

Sistema	Base	Dígitos
Binário	2	0, 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F



Regras de Conversão

- Para converter um número em qualquer sistema numérico para decimal, basta utilizar o polinômio de formatação, de acordo com a base





Exercício de fixação

Representar o número $(110101)_2$ em decimal usando o polinômio de formatação



Exercício de fixação

base 2

Representar o número $(110101)_2$ em decimal usando o polinômio de formatação

$$\begin{aligned} 1*2^5 &+ 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 \\ &32 + 16 + 4 + 1 \\ &(53)_{10} \end{aligned}$$



Sistemas de Conversão

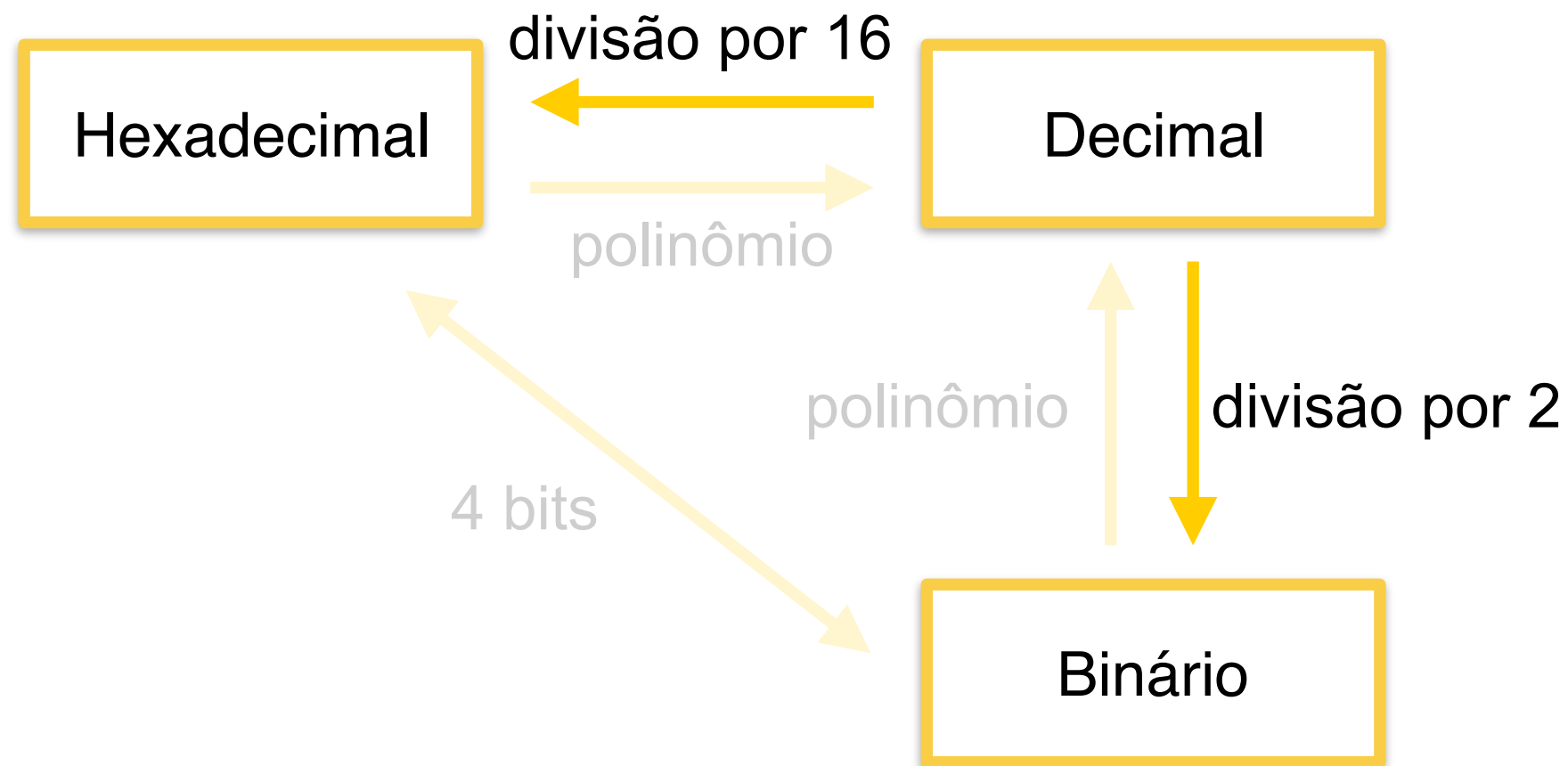
De qualquer base para o sistema de numeração decimal, vimos que basta utilizar o polinômio de formatação e somar os números de acordo com sua posição e a base.

Mas como representar $(345)_{10}$ no sistema binário?



Regras de Conversão

- Para converter um número em decimal para qualquer sistema numérico, utilizaremos o método das divisões sucessivas





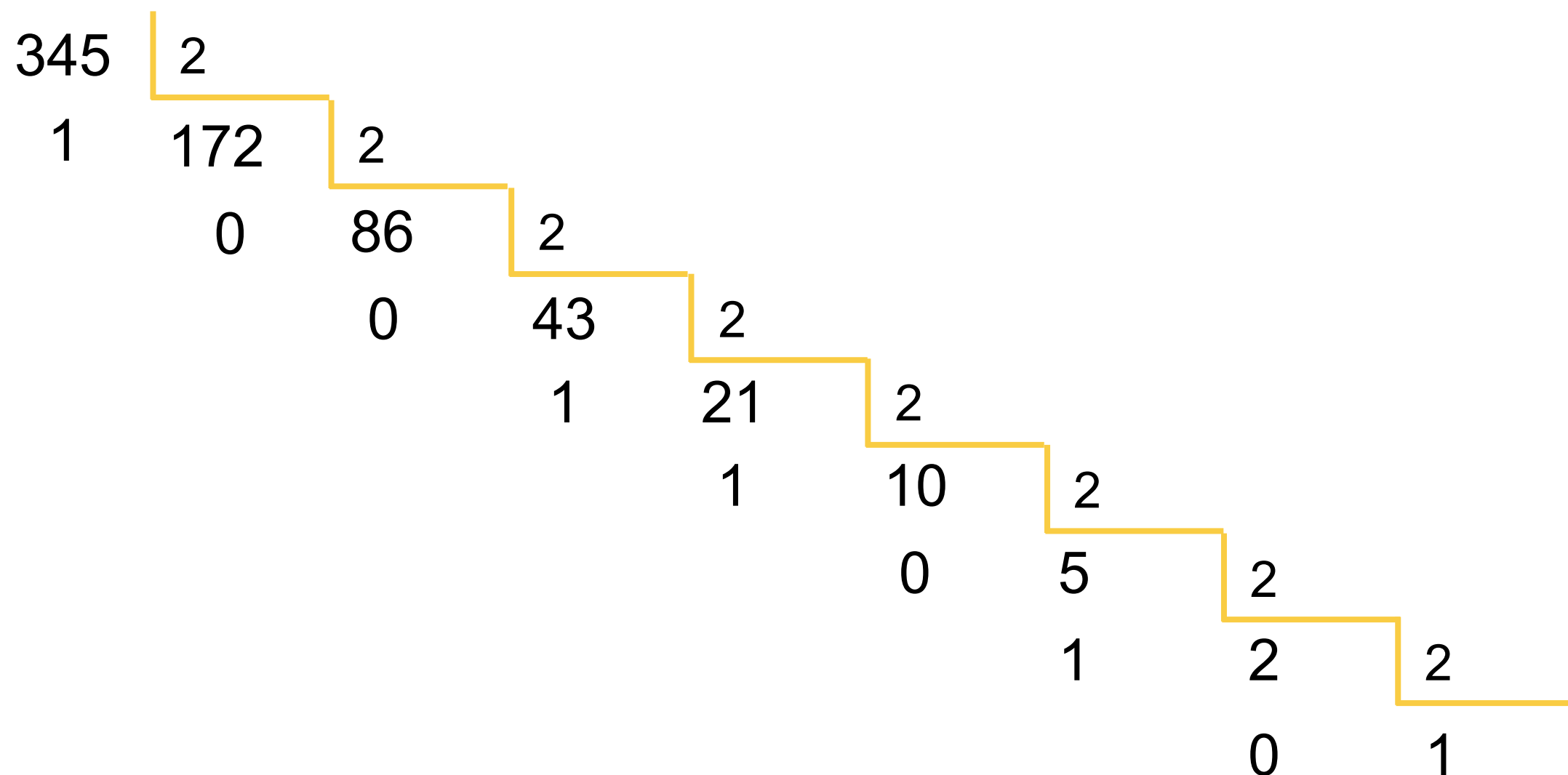
Sistema decimal para outro sistema

- **Passo 1:** Divisões sucessivas pela nova base
- **Atenção:** divisões exatas (com resto)

$$345 \begin{array}{l} | \\ 2 \end{array}$$



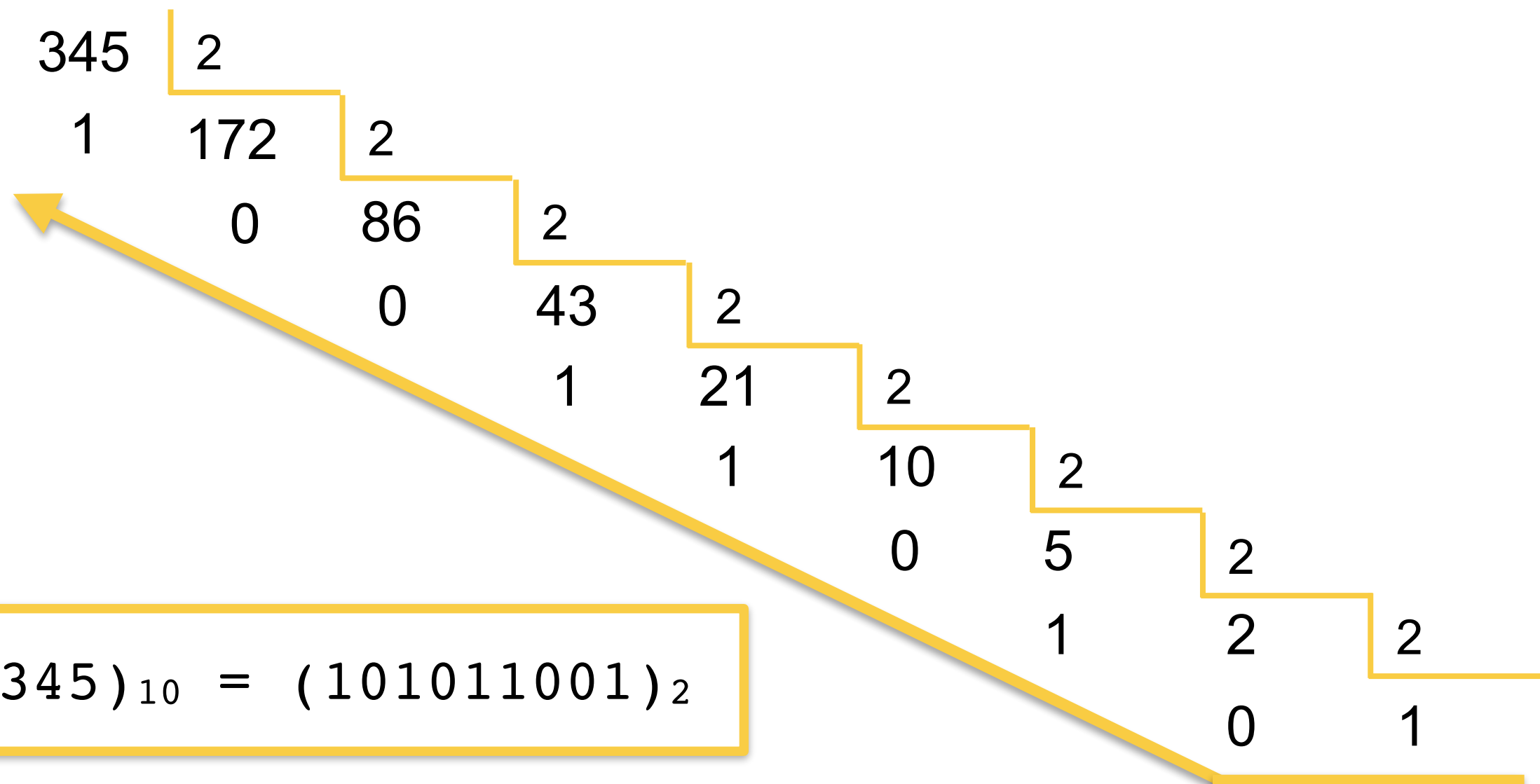
- **Passo 1:** Divisões sucessivas pela nova base
- Atenção: divisões exatas (com resto)





Sistema decimal para outro sistema

- **Passo 2:** Os restos das divisões, da direita para a esquerda, formarão o número em binário



$$(345)_{10} = (101011001)_2$$



Exercício de fixação

Representar o número $(298)_{10}$
para o sistema binário



Exercício de fixação

Representar o número $(169)_{10}$
para o sistema hexadecimal



Sistemas de Conversão

Entre os sistemas binário e hexadecimal, fazemos uma (des)compactação de bits

Por exemplo, o sistema hexadecimal representa números de 0 a F. Podemos representar esses números no sistema binário com 4 bits



Sistema binário para hexadecimal

- **Passo 1:** Agrupar o número em 4 bits, da direita para a esquerda
- **Atenção:** preencher com zero os dígitos restantes

0 0 0 1 0 0 1 0 1 1 1 1



Sistema binário para hexadecimal

- **Passo 2:** Calcular o equivalente em decimal
- Dica: será um número entre 0 e 15

0 0 0 1 0 0 1 0 1 1 1 1

1 2 15



Sistema binário para hexadecimal



Passo 3: Calcular o equivalente em hexadecimal

Sistema	Base	Dígitos
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

0 0 0 1 0 0 1 0 1 1 1 1

└───┬───┬───┘

1 2 F



Sistema binário para hexadecimal

Passo 3: Calcular o equivalente em hexadecimal

Sistema	Base	Dígitos
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

0 0 0 1 0 0 1 0 1 1 1 1

└───┴───┴───┘

1 2 F

$$(100101111)_2 = (12F)_{16}$$



Exercício de fixação

Representar o número $(10101110)_2$
para o sistema numérico hexadecimal



Sistema hexadecimal para binário



Passo 1: Calcular o equivalente em decimal

Sistema	Base	Dígitos
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

1B3

1 11 3



Sistema hexadecimal para binário

- **Passo 2:** Converter cada dígito para binário
- **Atenção:** cada dígito vai virar uma sequência de quatro bits. Completar com zeros quando preciso

1B3

1 11 3

0001 1011 0011



Sistema hexadecimal para binário

- **Passo 3:** Unir os resultados
- Eliminar zeros à esquerda, caso existam

1B3

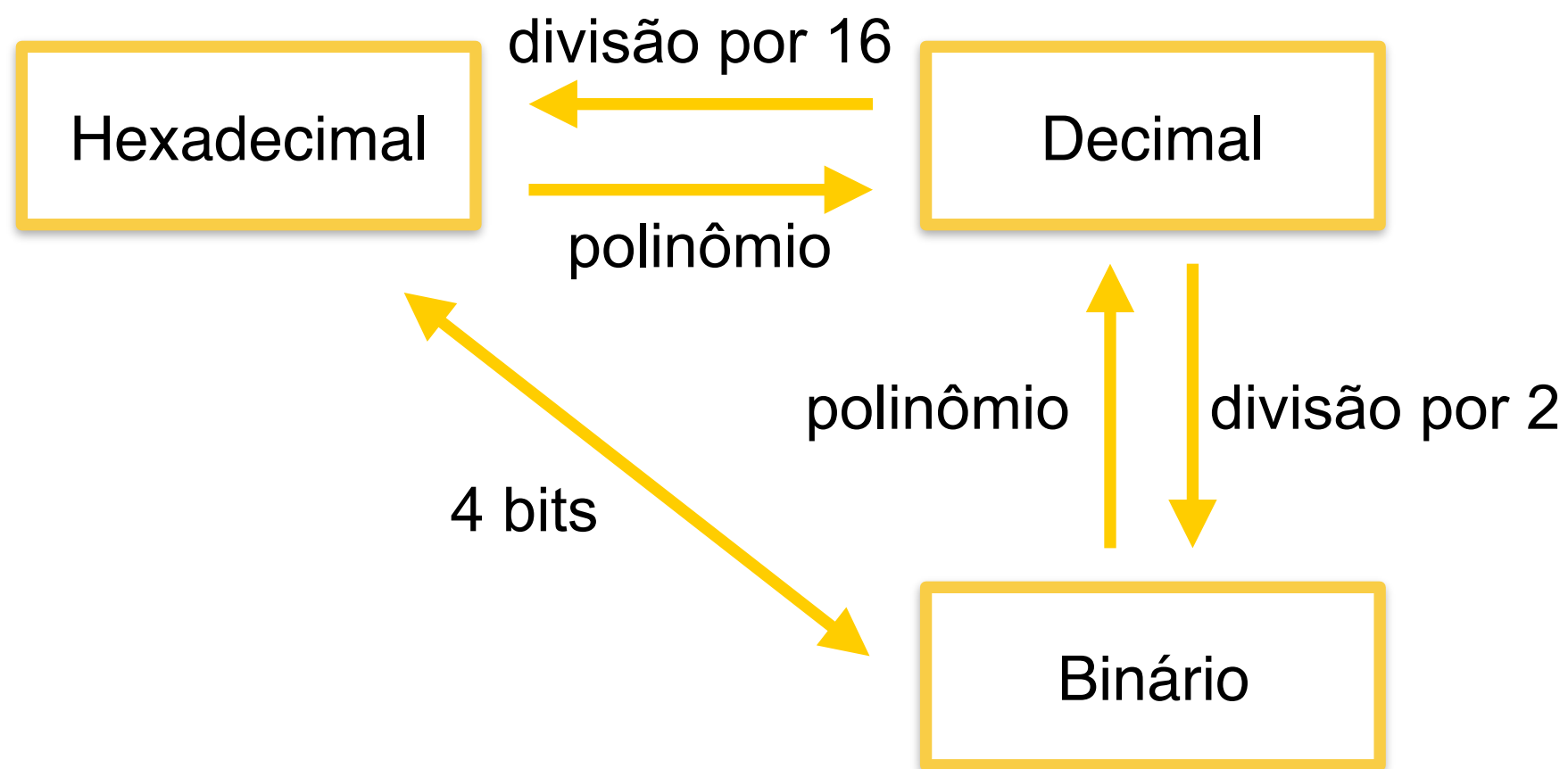
1 11 3

0001 1011 0011

$$(1B3)_{16} = (110110011)_2$$



Recapitulando...





Outras representações

Até agora vimos representações para números inteiros, em diferentes sistemas de numeração.

Mas também temos que representar letras e símbolos, além de números negativos e fracionados.



Representação de números negativos (1/2)

- Temos duas principais convenções
- **Representação de grandeza com sinal (1 bit)**



- Numa representação com 8 bits, temos:
- $(00101001)_2 = (?)_{10}$
- $(10101001)_2 = (?)_{10}$



Representação de números negativos (1/2)

- Temos duas principais convenções
- **Representação de grandeza com sinal (1 bit)**



- Numa representação com 8 bits, temos:
- $(00101001)_2 = (41)_{10}$
- $(10101001)_2 = (-41)_{10}$



Exercício de fixação

Meu computador utiliza 10 bits para representação de números em binário com sinal e magnitude.
Como represento $(-93)_{10}$ nessa máquina?



Representação de números negativos (1/2)

- Temos duas principais convenções
- **Representação de grandeza com sinal (1 bit)**
- **Desvantagens:**
 - Dupla representação para o zero
 - Requer computação extra para realizar operações de adição e subtração



Representação de números negativos (2/2)

- Temos duas principais convenções
- **Representação em complemento de 2**
- **Passo 1:** Manter os bits da direita para esquerda até a ocorrência do primeiro bit igual a 1

$$(00101001)_2 = (41)_{10}$$





Representação de números negativos (2/2)

- Temos duas principais convenções
- **Representação em complemento de 2**
- **Passo 1:** Manter os bits da direita para esquerda até a ocorrência do primeiro bit igual a 1
- **Passo 2:** Inverter os bits restantes

$$(00101001)_2 = (41)_{10}$$

$$(11010111)_2 = (-41)_{10}$$



Exercício de fixação

Meu computador utiliza 10 bits para representação de números em binário em complemento de 2.
Como represento $(-93)_{10}$ nessa máquina?



Representação de números negativos (2/2)

- Temos duas principais convenções
- **Representação em complemento de 2**
- **Vantagens:**
 - Apenas uma representação para o zero
 - Um circuito para adição e subtração

veremos na aula
de Aritmética
Binária



Exercício de fixação

Meu computador utiliza 8 bits para representação de números em binário. Como represento $(-56)_{10}$ nessa máquina:

- a) com grandeza de sinal?
- b) com complemento de 2?



Limite de representação (3 bits)

Decimal	Sinal e Magnitude	Compl. de 2
-4	—	100
-3	111	101
-2	110	110
-1	101	111
0	000 / 100	000
1	001	001
2	010	010
3	011	011



Limite de representação (4 bits)

Decimal	Binário com sinal	Compl. de 2
-8	—	1000
-7	1111	1001
-6	1110	1010
-5	1101	1011
-4	1100	1100
-3	1011	1101
-2	1010	1110
-1	1001	1111
0	0000 / 1000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111



Limite de representação (n bits)

- Com 3 bits, conseguimos representar números:
 - de -3 a 3 com sinal e magnitude
 - de -4 a 3 com complemento de 2
- Com 4 bits, conseguimos representar números:
 - de -7 a 7 com sinal e magnitude
 - de -8 a 7 com complemento de 2
- Podemos generalizar para n bits?



Exercício de fixação

Num sistema de representação binária com n bits, qual o limite de representação utilizando complemento de 2?

de $-(2^{n-1}-1)$ a $(2^{n-1}-1)$ em sinal e magnitude
de $-(2^{n-1})$ a $(2^{n-1}-1)$ em complemento de 2



Representação de símbolos

Para permitir que impressoras pudessem ser utilizadas por diferentes marcas de computadores, os fabricantes desde muito cedo decidiram criar uma codificação padrão para caracteres, baseadas em bits



Representação de símbolos (2/3)

- **ASCII** (American Standard Code for Information Interchange)
- **7 bits**, equivalente a **128** símbolos
- Computadores costumam representar uma versão estendida (8 bits), para permitir letras acentuadas



Representação de símbolos — ASCII

Decimal	Hexadecimal	ASCII
2	02	STX - Start of text
10	0A	LF - Line Feed
32	20	SP - Space
40	28	(- Left Parenthesis
41	29) - Right Parenthesis
97	61	a
65	41	A



Representação de símbolos (3/3)

- **UTF-8** (Unicode Transformation Format)
- **um a quatro bytes**, equivalente a mais de **1mi** de símbolos
- Compatibilidade com ASCII (primeiro byte)
- Codificação dominante na Web desde 2009



Representação de símbolos — UTF-8

Unicode point	Hexadecimal	UTF-8
U+0002	02	STX - Start of text
U+000A	0A	LF - Line Feed
U+0020	20	SP - Space
U+0028	28	(- Left Parenthesis
U+0029	29) - Right Parenthesis
U+0061	61	a
U+0041	41	A
U+0649	D989	س
U+1F602	F09F9882	😂



Bibliografia

TANENBAUM, Andrew S.; AUSTIN, Todd. **Organização estruturada de computadores**. 6 ed. São Paulo. Pearson Education do Brasil 2013 624 p. ISBN: 9788581435398.
(Apêndice A, disponível no Moodle)



Próxima Aula



Exercícios em Sala