Il sistema binario

Liceo G.B. Brocchi
Classi prime Scientifico – opzione scienze applicate
Bassano del Grappa, Ottobre 2022
Prof. Giovanni Mazzocchin

Il sistema decimale

- Il sistema che utilizziamo tutti i giorni si chiama **decimale**
- Utilizza 10 *cifre* (o *simboli*): {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
  - NB: da 0 a 10-1 = 9
- È un sistema **posizionale** perché il valore di una cifra dipende dalla sua posizione all'interno del numero (si dice anche dal suo **peso**)

$$9\quad 2\quad 6\quad 4_{\text{dec}}$$

9 migliaia **2 centinaia** **6 decine** **4 unità**

Il sistema decimale

- Unità, decine, centinaia, migliaia etc… sono i **pesi** (weights) delle cifre
- Possiamo usare le potenze di 10 al posto dei termini «unità», «decina» etc…

$$9\ 2\ 6\ 4_{dec}$$

9 * 10^3   **2 * 10^2**   **6 * 10^1**   **4 * 10^0**

Il simbolo ^ sta per «elevamento a potenza»

Il sistema decimale

- Immaginate di avere a disposizione una macchina elettronica che memorizza i numeri in formato decimale
- Dato un numero **x**, la macchina riserva lo spazio per **1** cifra decimale
- Quanti e quali numeri può rappresentare la macchina?

Valori possibili in memoria per x:

0
1
2
3
4
5
6
7
8
9

Se x = 9, la calcolatrice riesce a rappresentare x + 1?

Il sistema decimale

- Se x = 9, una calcolatrice che riserva solo uno spazio per i numeri non può calcolare x + 1
- Si dice che il calcolo x + 1 ha causato un **overflow** («traboccamento»)
- **Serve un altro spazio**, che qui visualizziamo a sinistra del precedente

Valori possibili in memoria per x:

00
01
02
03
04
05
06
07
08
09
10
11
20
20
99

Se x = 99, la calcolatrice riesce a rappresentare x + 1?

Il sistema decimale

- Se x = 99, una calcolatrice che riserva solo uno spazio per i numeri non può calcolare x + 1
- Si dice che il calcolo x + 1 ha causato un **overflow** («traboccamento»)
- **Serve un altro spazio**, che qui visualizziamo a sinistra del precedente

Valori possibili in memoria per x:

000
001
002
090
099
100
200
300
999

and so on…

Il sistema decimale

- Ipotizziamo di avere 3 spazi, ciascuno dei quali rappresentante una cifra decimale

Valori possibili in memoria per x:
000, 001, 002, 003, … 009
10, 11, 12…19
20, 22… 29
30, 31… 39
…
90, 91… 99
100, 101, 102… 109
110, 111… 119
990, 991, 992, 993, 994, 995, 996, 997, 998, 999

la nostra macchina non può più andare avanti

Procedura automatica («algoritmo») per scrivere i numeri in ordine:

Ripetere le seguenti istruzioni fintantoché ci sono ancora spazi non traboccati:

1. si fa traboccare lo spazio più a destra che non è ancora traboccato
2. al traboccamento, si porta a 0 lo spazio traboccato e si inserisce 1 nello spazio a sinistra di quello traboccato

Il sistema binario

- Uno dei sistemi più utilizzati in informatica ed elettronica è il sistema **binario**
- Utilizza 2 *cifre* (o *simboli*): {0, 1}
  - NB: da 0 a 2-1 = 1
- È un sistema **posizionale** perché il valore di una cifra dipende dalla sua posizione all'interno del numero (si dice anche dal suo **peso**)

$$1\quad 1\quad 0\quad 1_{bin}$$

1 * 2^3 **1 * 2^2** **0 * 2^1** **1 * 2^0**

Il sistema binario

- Immaginate di avere a disposizione una macchina elettronica che memorizza i numeri in formato binario
- Dato un numero **x**, la macchina riserva lo spazio per **1** cifra binaria
- Quanti e quali numeri può rappresentare la macchina?

Valori possibili in memoria per x:

0

1

Se x = 1, la calcolatrice riesce a rappresentare x + 1?

Il sistema binario

- Se x = 1, una calcolatrice che riserva solo uno spazio per i numeri non può calcolare x + 1
- Si dice che il calcolo x + 1 ha causato un **overflow** («traboccamento»)
- **Serve un altro spazio**, che qui visualizziamo a sinistra del precedente

Valori possibili in memoria per x:

00
01
10
11

Se x = 11, la calcolatrice riesce a rappresentare x + 1?

Il sistema binario

- Se x = 11, una calcolatrice che riserva solo uno spazio per i numeri non può calcolare x + 1
- Si dice che il calcolo x + 1 ha causato un **overflow** («traboccamento»)
- **Serve un altro spazio**, che qui visualizziamo a sinistra del precedente

Possibili valori in memoria per x

00
01
10
11
100
101
110
111

Il sistema binario

Procedura automatica («algoritmo») per scrivere i numeri in ordine:

Ripetere le seguenti istruzioni fintantoché ci sono ancora spazi non traboccati:

1. si fa traboccare lo spazio più a destra che non è ancora traboccato
2. al traboccamento, si porta a 0 lo spazio traboccato e si inserisce 1 nello spazio a sinistra di quello traboccato

Esempio con 4 spazi, 1 cifra binaria per ogni spazio

00
01
10
11
100
101
110
111
1000
1001
1010
1011
1100
1101
1110
1111

Abbiamo rappresentato 16 = 2^4 numeri.
Da 0_{dec} a 15_{dec}

Il sistema binario

- Vogliamo rappresentare il numero 90_{dec} in binario
- Come fare? **NB**: tutti i numeri in questa slide sono scritti nel sistema decimale
- Proviamo a rappresentare 90 come somma di potenze di 2 (base del sistema binario)
  - 90_{dec} =

$$
\begin{aligned}
&64 + 26 = \\
&2\text{^}6 + 26 = \\
&2\text{^}6 + 16 + 10 = \\
&2\text{^}6 + 2\text{^}4 + 10 = \\
&2\text{^}6 + 2\text{^}4 + 8 + 2 = \\
&2\text{^}6 + 2\text{^}4 + 2\text{^}3 + 2 = \\
&2\text{^}\mathbf{6} + 2\text{^}\mathbf{4} + 2\text{^}\mathbf{3} + 2\text{^}\mathbf{1}
\end{aligned}
$$

$$\mathbf{1\,0\,1\,1\,0\,1\,0}_{\mathbf{bin}}$$

Le cifre in posizione 0, 2, 5 sono a zero perché non sono presenti nello sviluppo di 90_{dec} come somma di potenze di 2

Il sistema binario

- Il calcolo della slide precedente è stato effettuato a colpi di «buon senso»

- In informatica abbiamo bisogno di procedure completamente automatizzabili, dato che le macchine elettroniche non conoscono il *buon senso*

- Dato un numero naturale *n,* questo può essere:
    1. una potenza di 2. In questo caso la divisione intera di n per la potenza di 2 a lui più vicina ha. Esempio: n = 64, quindi n = 2^6. 64 / 64 = 1, con resto 0;
    2. diverso da una potenza di 2. In questo caso dobbiamo trovare una somma di potenze di 2 che uguagli il n.

Il sistema binario

- **n = 64**
  - Dobbiamo capire in modo automatico qual è la potenza di 2 più vicina a n
  - Vogliamo cioè scrivere 64 come 2^x, senza sapere a priori quanto vale x
  - Lo dividiamo per 2 finché non otteniamo 1!
    - 64 / 2 = 32
    - 32 / 2 = 16
    - 16 / 2 = 8
    - 8 / 2 = 4
    - 4 / 2 = 2
    - 2 / 2 = 1

6 divisioni

64 = 2 * 2 * 2 * 2 * 2 * 2
64 / 2 = 2 * 2 * 2 * 2 * 2
64 / (2*2) = 2 * 2 * 2 * 2
64 / (2*2*2) = 2 * 2 * 2
64 / (2*2*2*2) = 2 * 2
64 / (2*2*2*2*2) = 2
64 / (2*2*2*2*2*2) = 1

64 / (2^6) = 1 significa 64 = 2^6

Il sistema binario

- Abbiamo scoperto in modo *algoritmico* (come farebbe una macchina non pensante) che 64 = 2^6
- Ora, sappiamo che la cifra binaria in posizione 6 (partendo da 0) ha peso 2^6:
  - Quindi $64_{dec} = 1\ 0\ 0\ 0\ 0\ 0\ 0_{bin}$

0 * 2^5

1 * 2^6

Il sistema binario

- **n = 91**
  - Dobbiamo capire in modo automatico qual è la potenza di 2 più vicina a n, che in questo caso non è una potenza di 2
  - Vogliamo cioè scrivere 91 come **2^x + y**, senza sapere a priori quanto valgono x e y
  - Lo dividiamo per 2 finché non ottieniamo 1!
    - 91 / 2 = 45 (resto = 1)
    - 45 / 2 = 22 (resto = 1)
    - 22 / 2 = 11 (resto = 0)
    - 11 / 2 = 5 (resto = 1)
    - 5 / 2 = 2 (resto = 1)
    - 2 / 2 = 1 (resto = 0)

6 divisioni. Quindi la potenza di 2 più vicina a 91 è 2^6

Il sistema binario

- **n = 91**

- $91 = 2^6 + y = 1\,0\,0\,0\,0\,0\,0_{bin} + y$

- quindi $y = 91 - 2^6 = 27$

- Ora dobbiamo esprimere anche 27 come somma di potenze di 2!
  - 27 / 2 = 13 (resto = 1)
  - 13 / 2 = 6 (resto = 1)
  - 6 / 2 = 3 (resto = 0)
  - 3 / 2 = 1 (resto = 1)

4 divisioni. Quindi la potenza di 2 più vicina a 27 è 2^4

Il sistema binario

- **n = 91**

- $91 = 2^6 + 2^4 + y = 1\ 0\ 0\ 0\ 0\ 0\ 0_{bin} + 1\ 0\ 0\ 0\ 0_{bin} + y$

- Quindi $y = 91 - 2^6 - 2^4 = 11$

- Ora dobbiamo esprimere anche 11 come somma di potenze di 2!
  - 11 / 2 = 5 (resto = 1)
  - 5 / 2 = 2 (resto = 1)
  - 2 / 2 = 1 (resto = 0)

3 divisioni. Quindi la potenza di 2 più vicina a 1 è 2^3

Il sistema binario

- **n = 91**
- 91 = 2^6 + 2^4 + 2^3 + y = $1\,0\,0\,0\,0\,0\,0_{bin}$ + $1\,0\,0\,0\,0_{bin}$ + $1\,0\,0\,0_{bin}$ + y
- Quindi y = 91 – 2^6 – 2^4 – 2^3 = 3
- Ora dobbiamo esprimere anche 3 come somma di potenze di 2!
  - 3 / 2 = 1 (resto = 1)

1 divisione. Quindi la potenza di 2 più vicina a 1 è 2^1

Il sistema binario

- **n = 91**
- $91 = 2^6 + 2^4 + 2^3 + 2^1 + y = 1\ 0\ 0\ 0\ 0\ 0\ 0_{bin} + 1\ 0\ 0\ 0\ 0_{bin} + 1\ 0\ 0\ 0_{bin} + 10_{bin} + y$
- Quindi $y = 91 - 2^6 - 2^4 - 2^3 - 2^1 = 1$
- 1 è già una potenza di 2 (2^0).

$91 = 2^6 + 2^4 + 2^3 + 2^1 + 2^0 = 1\ 0\ 0\ 0\ 0\ 0\ 0_{bin} + 1\ 0\ 0\ 0\ 0_{bin} + 1\ 0\ 0\ 0_{bin} + 10_{bin} + 1_{bin}$

Il sistema binario

1	0	0	0	0	0	0	**+**
		1	0	0	0	0	**+**
			1	0	0	0	**+**
					1	0	**+**
						1	**=**
1	**0**	**1**	**1**	**0**	**1**	**1**	

Il sistema binario

- 91 / 2 = 45 (resto = **1**)
- 45 / 2 = 22 (resto = **1**)
- 22 / 2 = 11 (resto = **0**)
- 11 / 2 = 5 (resto = **1**)
- 5 / 2 = 2 (resto = **1**)
- 2 / 2 = 1 (resto = **0**)
- 1 / 2 = 0 (resto = **1**)

Leggendo i resti al contrario viene fuori il numero che abbiamo calcolato prima in modo più macchinoso.
NB: sicuramente il resto di una divisione intera per 2 può essere solo 0 oppure 1

Addizione in colonna decimale

$$\begin{array}{ccc} 1 & 9 & + \\ & 8 & \\ \hline 2 & 7 & \end{array}$$

Descrivere in modo *algoritmico* la procedura detta addizione decimale in colonna

Addizione in colonna decimale

$$\begin{array}{ccc} & 9 & 9 & + \\ & 9 & 9 & \\ \hline 1 & 9 & 8 & \end{array}$$

1. 9 + 9 = 18. Traboccamento e riporto;
2. 9 + 9 = 18, 18 + ***1*** = 19. Traboccamento e riporto;
3. 1 + 0 = 1;
4. Fine della procedura.

Addizione in colonna decimale

$$\begin{array}{cccccc} & 9 & 9 & 9 & + \\ & 9 & 9 & 9 & \\ \hline 1 & 9 & 9 & 8 & \end{array}$$

1. 9 + 9 = 18. Traboccamento e riporto.
2. 9 + 9 = 18, 18 + ***1*** = 19. Traboccamento e riporto;
3. 9 + 9 = 18, 18 + ***1*** = 19. Traboccamento e riporto
4. 0 + 1 = 1;
5. Fine della procedura.

NB: qualsiasi sia il sistema di numerazione, la cifra da riportare a sinistra in caso di traboccamento è sempre 1.

Addizione in colonna tra numeri nel sistema binario

$$
\begin{array}{cc}
1 & 0 \quad + \\
 & 1 \\
\hline
1 & 1
\end{array}
$$

1. 0 + 1 = 1;
2. Secondo spazio 1 + 0 = 1;
3. Fine della procedura.

Addizione in colonna tra numeri nel sistema binario

	1	1	+
	1	1	
1	1	0	

$3_{dec} + 3_{dec} = 6_{dec}$

1. 1 + 1 = 10. Traboccamento e riporto;
2. 1 + 1 = 10. 10 + ***1*** = *11*. Traboccamento e riporto;
3. 0 + 1 = 1;
4. Fine della procedura.

Addizione in colonna tra numeri in sistema binario

	1	1	1	+
	1	1	1	
1	1	1	0	

$7_{dec} + 7_{dec} = 14_{dec}$

1. 1 + 1 = 10. Traboccamento e riporto;
2. 1 + 1 = 10, 10 + ***1*** = 11. Traboccamento e riporto;
3. 1 + 1 = 10, 10 + 1 = 11. Traboccamento e riporto;
4. 0 + 1 = 1;
5. Fine della procedura.

Sottrazione in colonna tra numeri nel sistema decimale

2	5	-
	9	
1	6	

prima, seconda, terza, quarta etc.. colonna: si conta a partire da destra

1. 5 – 9 < 0. Necessità di un **prestito** dalla seconda colonna (**borrow**);
2. Il 2 della seconda colonna del sottraendo diventa 1, il 5 della prima colonna diventa 15;
3. 15 – 9 = 6. Prima colonna completata;
4. 1 – 0 = 1. Seconda colonna completata;
5. Fine della procedura.

Sottrazione in colonna tra numeri nel sistema decimale

1	0	5	-
	8	9	
	1	6	

1. 5 – 9 < 0. Necessità di un **prestito** da sinistra (**borrow**);
2. La seconda colonna del sottraendo non può prestare niente;
3. La terza colonna presta 1 alla seconda e diventa 0;
4. La seconda colonna diventa 10;
5. La seconda colonna presta 1 alla prima e diventa 9;
6. Il 5 della prima colonna del sottraendo diventa 15;
7. 15 – 9 = 6. Prima colonna completata;
8. 9 – 8 = 1. Seconda colonna completata;
9. Fine della procedura.

Sottrazione in colonna tra numeri nel sistema decimale

$$
\begin{array}{cccc}
0 & {}^{1}0 & 5 & - \\
 & 8 & 9 & \\
\hline
 & 1 & 6 &
\end{array}
$$

1. 5 – 9 < 0. Necessità di un **prestito** da sinistra (**borrow**);
2. La seconda colonna del sottraendo non può prestare niente;
3. La terza colonna presta 1 alla seconda e diventa 0;
4. La seconda colonna diventa 10;
5. La seconda colonna presta 1 alla prima e diventa 9
6. Il 5 della prima colonna del sottraendo diventa 15;
7. 15 – 9 = 6. Prima colonna completata;
8. 9 – 8 = 1. Seconda colonna completata;
9. Fine della procedura.

Sottrazione in colonna tra numeri nel sistema decimale

$$
\begin{array}{cccc}
0 & 9 & {}^{1}5 & - \\
 & 8 & 9 & \\
\hline
 & 1 & 6 &
\end{array}
$$

1. 5 – 9 < 0. Necessità di un **prestito** da sinistra (**borrow**);
2. La seconda colonna del sottraendo non può prestare niente;
3. La terza colonna presta 1 alla seconda e diventa 0;
4. La seconda colonna diventa 10;
5. La seconda colonna presta 1 alla prima e diventa 9
6. Il 5 della prima colonna del sottraendo diventa 15;
7. 15 – 9 = 6. Prima colonna completata;
8. 9 – 8 = 1. Seconda colonna completata;
9. Fine della procedura.

Sottrazione in colonna tra numeri nel sistema decimale

9	0	0	5	-
	9	9	9	

Sottrazione in colonna tra numeri nel sistema decimale

8	$^{1}0$	0	5	-
	9	9	9	

Sottrazione in colonna tra numeri nel sistema decimale

$$
\begin{array}{cccccc}
8 & 9 & {}^{1}0 & 5 & - \\
 & 9 & 9 & 9 & \\
\hline
\end{array}
$$

Sottrazione in colonna tra numeri nel sistema decimale

$$
\begin{array}{cccccc}
8 & 9 & 9 & {}^{1}5 & - \\
 & 9 & 9 & 9 & \\
\hline
8 & 0 & 0 & 6 &
\end{array}
$$

Sottrazione in colonna tra numeri nel sistema binario

1	1	0	-
		1	

$6_{dec} - 1_{dec} = 5_{dec}$

Sottrazione in colonna tra numeri nel sistema binario

$$
\begin{array}{cccc}
 & & {}^{1} & \\
1 & 0 & 0 & - \\
 & & 1 & \\
\hline
1 & 0 & 1 &
\end{array}
$$

$6_{dec} - 1_{dec} = 5_{dec}$

Sottrazione in colonna tra numeri nel sistema binario

1	0	0	0	0	-
	1	1	1	1	

$16_{dec} - 15_{dec} = 1_{dec}$

L'occhio esperto vede subito che il risultato è 1.
E' la differenza tra 2^n e (2^n - 1)

Sottrazione in colonna tra numeri nel sistema binario

$$
\begin{array}{cccccc}
0 & {}^{1}0 & 0 & 0 & 0 & - \\
 & 1 & 1 & 1 & 1 & \\
\hline
\end{array}
$$

Sottrazione in colonna tra numeri nel sistema binario

0	1	[1]0	0	0	-
	1	1	1	1	

Sottrazione in colonna tra numeri nel sistema binario

$$
\begin{array}{cccccc}
0 & 1 & 1 & {}^{1}0 & 0 & - \\
  & 1 & 1 & 1 & 1 & \\
\hline
\end{array}
$$

Sottrazione in colonna tra numeri nel sistema binario

$$
\begin{array}{cccccc}
0 & 1 & 1 & 1 & {}^{1}0 & - \\
 & 1 & 1 & 1 & 1 & \\
\hline
0 & 0 & 0 & 0 & 1 &
\end{array}
$$

$16_{dec} - 15_{dec} = 1_{dec}$

Sottrazione in colonna tra numeri nel sistema binario

1	0	0	0	0	-
	1	1	1	1	

2^4

Un numero binario composto da n 1 è uguale a 2^n - 1

2^4 - (2^4 - 1) = 1

Moltiplicazione nel sistema binario

$$
\begin{array}{cccccccc}
 & & 1 & 1 & 0 & 0 & 0 & x \\
 & & & & & 1 & 1 & \\
\hline
 & & 1 & 1 & 0 & 0 & 0 & + \\
 & 1 & 1 & 0 & 0 & 0 & & \\
\hline
1 & 0 & 0 & 1 & 0 & 0 & 0 &
\end{array}
$$

24_{dec} * $3*_{dec}$ = 72_{dec}

Divisione nel sistema decimale

- Vediamo la divisione come **sottrazione ripetuta**:

  20 / 6 = 3, resto 2

20 - 6 = 14

14 - 6 = 8

8 - 6 = 2

2 - 6 → STOP

Il quoziente è il numero di volte che siamo riusciti a sottrarre senza oltrepassare lo zero

Il resto è il sottraendo della sottrazione che non siamo riusciti a fare

Divisione nel sistema decimale

- Vediamo la divisione come **sottrazione ripetuta**:
  28 / 7 = 4, resto 0

28 - 7 = 21
21 - 7 = 14
14 - 7 = 7
7 - 7 = 0

STOP

Il quoziente è il numero di volte che siamo riusciti a sottrarre senza oltrepassare lo zero

Divisione nel sistema binario

- Vediamo la divisione come **sottrazione ripetuta**:

  10100 / 110

1	0	1	0	0	-
		1	1	0	
	1	1	1	0	-
		1	1	0	
	1	0	0	0	-
		1	1	0	
			1	0	-
		1	0	0	

→ **STOP**