

Introduzione all'Architettura degli Elaboratori

Sommario

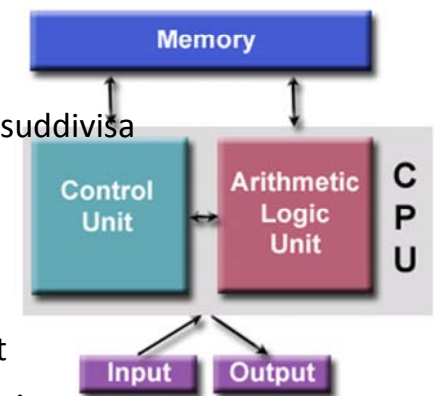
- Macchina di von Neumann
- Esecuzione dei programmi
- Rappresentazione dei dati
 - Dati numerici
 - Dati alfabetici

Il Modello di von Neumann (1)

- L'architettura di un computer è basata sul modello proposto da Janos (John) von Neumann alla metà degli anni '40 del secolo scorso

Il Modello di von Neumann (2)

- Prevede 3 entità logiche:
 - La memoria
 - La Central Processing Unit, suddivisa in
 - Control Unit
 - Arithmetic Logic Unit
 - I dispositivi di Input/Output
- Le informazioni (dati) viaggiano tra le componenti mediante un bus



Central Processing Unit (1)

- È la componente che acquisisce, interpreta ed esegue le istruzioni dei programmi
- Si compone a sua volta di
 - **Control Unit**, responsabile del prelievo e della decodifica delle istruzioni e dell'invio dei segnali di controllo
 - **Arithmetic Logic Unit**, per l'esecuzione delle operazioni aritmetiche e logiche
 - Alcune varianti del modello prevedono anche un **clock**

Central Processing Unit (2)

- Da un punto di vista operativo, la CPU fa uso di alcuni **registri**:
 - Elementi di memoria i cui valori possono essere acceduti in lettura e scrittura molto velocemente
 - Registro Istruzione Corrente (**CIR**) contiene l'istruzione in corso di esecuzione
 - Contatore di Programma (**PC**) contiene l'indirizzo della prossima istruzione del programma in esecuzione

Memory (1)

- È la memoria centrale, di lavoro
 - da non confondersi con la memoria di massa in cui vengono immagazzinati dati e programmi quando non in uso
- Contiene gli elementi che il computer sta usando nella elaborazione corrente, e precisamente
 - Le istruzioni del programma in corso di esecuzione
 - I dati necessari all'esecuzione di quel programma

Memory (2)

- Si può immaginare la memoria come costituita da tante celle, ognuna identificata univocamente da un proprio indirizzo

I/O

- Sono i dispositivi con cui rispettivamente
 - vengono forniti dati e programmi al computer
 - vengono prodotti dal computer i risultati dell'elaborazione

Elementi di supporto

- La macchina di von Neumann è una macchina astratta, un modello per realizzare macchine reali
- Per questo non vengono enfatizzati ulteriori elementi logici che pure sono necessari, come a esempio
 - i già citati bus e clock di sistema
 - le memorie di massa
 - le interfacce di rete
 - ...

Esecuzione dei Programmi (1)

- La macchina di von Neumann è in grado di eseguire programmi espressi in un opportuno **linguaggio macchina**
 - Per le macchine reali il linguaggio macchina è codificato secondo codici binari o esadecimali

Esecuzione dei Programmi (2)

- Si suppone che il programma da eseguire sia caricato nella Memory prima dell'esecuzione
 - da qualche memoria in cui è stato precedentemente registrato
 - fornito in input dal programmatore
 - ...
- Si suppone inoltre che il programma sia suddiviso logicamente in due parti
 - L'insieme di istruzioni che devono essere eseguite
 - I dati (di input/output e di supporto calcolati e temporanei) su cui le istruzioni operano

Il Ciclo Fetch-Execute (1)

- L'esecuzione del programma avviene ripetendo iterativamente le fasi di
 - **Acquisizione (fetch)** dell'istruzione da eseguire dalla Memory
 - **Interpretazione (decodifica)** dell'istruzione
 - **Esecuzione** dell'istruzione

Il Ciclo Fetch-Execute (2)

- Il contenuto del PC viene caricato nel CIR
- Il PC viene aggiornato con l'indirizzo dell'istruzione successiva
- L'istruzione del CIR viene decodificata ed eseguita
 - Eventualmente accedendo a dati

Rappresentazione dei Dati

- Tutte le informazioni (dati e istruzioni) in un computer sono rappresentate in forma binaria
 - Cioè come sequenza finita di simboli '0' e '1'
- L'unità di informazione è il **bit (binary digit)**
 - un bit può assumere solo i valori 0 oppure 1
- Una sequenza di 8 bit costituisce un **byte**
 - Un byte può assumere uno tra i 2^8 valori
00000000, 00000001, 00000010, ..., 11111111
- Con **word** si intende N byte
 - N dipendente dallo specifico contesto

Rappresentazione di Dati Numerici (1)

- Data la finitezza della macchina computer, tutte le informazioni che in esso si possono rappresentare sono necessariamente **finite**
- Quando, a esempio, si parlerà di
 - “insieme dei numeri interi” si intenderà un'approssimazione finita di \mathbb{N}
 - “insieme dei numeri reali” si intenderà un'approssimazione finita di \mathbb{R}

Rappresentazione di Dati Numerici (2)

- Siamo abituati a pensare (e usare) i numeri interi in forma **decimale** e **posizionale**
 - Decimale, perché espressi in base 10, facendo uso di dieci simboli ('0', '1', '2', '3', '4', '5', '6', '7', '8', '9')
 - Posizionale perché la posizione in cui compare un simbolo è fondamentale per la sua interpretazione (il numero 12 e il numero 21 sono rappresentati dagli stessi simboli '1' e '2', ma disposti in posizioni diverse)

Rappresentazione di Dati Numerici (3)

- Sia **N** un generico numero intero: in forma decimale e posizionale è rappresentato come la sequenza di **n** simboli (cifre decimali)

$$a_n a_{n-1} a_{n-2} \dots a_2 a_1 a_0$$

- dove le a_i sono le diverse cifre, con a_n cifra più significativa, e a_0 cifra meno significativa

- Se **N** è espresso in base **b**, allora vale

$$N_b = a_n * b^n + a_{n-1} * b^{n-1} + a_{n-2} * b^{n-2} + \dots + a_2 * b^2 + a_1 * b^1 + a_0 * b^0$$

Rappresentazione di Dati Numerici (4)

- Esempi
 - $485_{10} = (4 * 10^2 + 8 * 10^1 + 5 * 10^0)_{10}$
 - $3642_8 = (3 * 8^3 + 6 * 8^2 + 4 * 8^1 + 2 * 8^0)_{10} = (3 * 512 + 6 * 64 + 4 * 8 + 2 * 1)_{10} = (1536 + 384 + 32 + 2)_{10} = 1954_{10}$
 - $9C3_{16} = (9 * 16^2 + 12 * 16^1 + 3 * 16^0)_{10} = (9 * 256 + 12 * 16 + 3 * 1)_{10} = (2304 + 192 + 3)_{10} = 2499_{10}$
 - $10011001_2 = (1 * 2^7 + 0 * 2^6 + 0 * 2^5 + 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0)_{10} = (128 + 0 + 0 + 16 + 8 + 0 + 0 + 1)_{10} = 153_{10}$

Rappresentazione di Dati Numerici (5)

- Esempio di conversione di un numero decimale (342) in binario

342 : 2 = 171	con resto 0
171 : 2 = 85	con resto 1
85 : 2 = 42	con resto 1
42 : 2 = 21	con resto 0
21 : 2 = 10	con resto 1
10 : 2 = 5	con resto 0
5 : 2 = 2	con resto 1
2 : 2 = 1	con resto 0
1 : 2 = 0	con resto 1

diventa il bit meno significativo

$$342_{10} = 101010110_2$$

diventa il bit più significativo

Rappresentazione di Interi (1)

- Possiamo considerare due casi:
 - Interi senza segno - **unsigned integer**: si intendono solo i positivi
 - Interi con segno - **signed integer**: si intendono i positivi e i negativi
 - Immaginiamo che il computer possa utilizzare una word di N byte per rappresentare un integer (sia signed che unsigned): $N \text{ byte} = N * 8 \text{ bit} = W \text{ bit}$

Rappresentazione di Interi (2)

- Avendo a disposizione W bit è possibile rappresentare i valori binari da 0 a $2^W - 1$
- Nel caso si voglia rappresentare gli interi signed il bit più significativo viene utilizzato per indicare il segno
 - Convenzionalmente '0'='+' e '1'='-'
- In tal caso quindi con W bit si possono rappresentare gli interi compresi tra $-(2^{W-1}-1)$ e $+(2^{W-1}-1)$

Rappresentazione di Reali (1)

- Nell'ambito di un computer l'approssimazione di \mathbb{R} espressa da "numeri reali" indica più precisamente un'approssimazione (finita) dei numeri razionali
- Concettualmente, un numero "reale" è rappresentato dalla giustapposizione di due numeri
 - Se W bit sono disponibili, i primi W_i indicano la parte intera, i rimanenti W_f la parte frazionaria
- Tale rappresentazione è detta a **virgola fissa (fixed point)**

Rappresentazione di Reali (2)

- La modalità di rappresentazione più diffusa è quella a **virgola mobile (floating point)**
- Utilizza due valori
 - **Mantissa**, interpretato come numero frazionario tra -1 e +1
 - **Caratteristica**, usato come esponente
- Si basa sulla notazione esponenziale, secondo cui
$$r = m * b^c$$
 - dove r è il numero reale da rappresentare, m è la mantissa, b è la base di rappresentazione e c la caratteristica

Rappresentazione di Caratteri (1)

- Anche i caratteri sono rappresentati mediante codici binari
- Più in generale si parla di “simboli” per specificare oltre agli usuali caratteri alfabetici, anche i simboli che indicano
 - le cifre (decimali)
 - la punteggiatura
 - i simboli speciali (blank, carriage return, linefeed, ...)

Rappresentazione di Caratteri (2)

- Per poter codificare univocamente i caratteri è necessaria una corrispondenza biunivoca tra questi e un opportuno sottoinsieme degli interi
 - Standard ASCII (American Standard Code for Information Interchange)
 - UNICODE

Rappresentazione di Caratteri (3)

- Funzioni di trasferimento
 - ord(c): numero d'ordine del simbolo c nella tavola di codifica
 - chr(i): Simbolo il cui numero d'ordine è i
- Proprietà
 - $\text{ord}(\text{chr}(i))=i$ $\text{chr}(\text{ord}(c))=c$
 - SE $c_1 < c_2$ ALLORA $\text{ord}(c_1) < \text{ord}(c_2)$
- Relazione d'ordine totale
 - Coerente con i sottoinsiemi delle lettere e delle cifre

Tabella ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	€#32;	Space	64	40	100	€#64;	B	96	60	140	€#96;	a
1	1	001	SOH (start of heading)	33	21	041	€#33;	!	65	41	101	€#65;	A	97	61	141	€#97;	A
2	2	002	STX (start of text)	34	22	042	€#34;	"	66	42	102	€#66;	B	98	62	142	€#98;	b
3	3	003	ETX (end of text)	35	23	043	€#35;	#	67	43	103	€#67;	C	99	63	143	€#99;	c
4	4	004	EOT (end of transmission)	36	24	044	€#36;	\$	68	44	104	€#68;	D	100	64	144	€#100;	d
5	5	005	ENQ (enquiry)	37	25	045	€#37;	%	69	45	105	€#69;	E	101	65	145	€#101;	e
6	6	006	ACK (acknowledge)	38	26	046	€#38;	&	70	46	106	€#70;	F	102	66	146	€#102;	f
7	7	007	BEL (bell)	39	27	047	€#39;	'	71	47	107	€#71;	G	103	67	147	€#103;	g
8	8	010	BS (backspace)	40	28	050	€#40;	(72	48	110	€#72;	H	104	68	150	€#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	€#41;)	73	49	111	€#73;	I	105	69	151	€#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	€#42;	*	74	4A	112	€#74;	J	106	70	152	€#106;	j
11	B	013	VT (vertical tab)	43	2B	053	€#43;	+	75	4B	113	€#75;	K	107	71	153	€#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	€#44;	,	76	4C	114	€#76;	L	108	72	154	€#108;	l
13	D	015	CR (carriage return)	45	2D	055	€#45;	-	77	4D	115	€#77;	M	109	73	155	€#109;	m
14	E	016	SO (shift out)	46	2E	056	€#46;	.	78	4E	116	€#78;	N	110	74	156	€#110;	n
15	F	017	SI (shift in)	47	2F	057	€#47;	/	79	4F	117	€#79;	O	111	75	157	€#111;	o
16	10	020	DLE (data link escape)	48	30	060	€#48;	0	80	50	120	€#80;	P	112	76	160	€#112;	p
17	11	021	DC1 (device control 1)	49	31	061	€#49;	1	81	51	121	€#81;	Q	113	77	161	€#113;	q
18	12	022	DC2 (device control 2)	50	32	062	€#50;	2	82	52	122	€#82;	R	114	78	162	€#114;	r
19	13	023	DC3 (device control 3)	51	33	063	€#51;	3	83	53	123	€#83;	S	115	79	163	€#115;	s
20	14	024	DC4 (device control 4)	52	34	064	€#52;	4	84	54	124	€#84;	T	116	80	164	€#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	€#53;	5	85	55	125	€#85;	U	117	81	165	€#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	€#54;	6	86	56	126	€#86;	V	118	82	166	€#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	€#55;	7	87	57	127	€#87;	W	119	83	167	€#119;	w
24	18	030	CAN (cancel)	56	38	070	€#56;	8	88	58	130	€#88;	X	120	78	170	€#120;	x
25	19	031	EM (end of medium)	57	39	071	€#57;	9	89	59	131	€#89;	Y	121	79	171	€#121;	y
26	1A	032	SUB (substitute)	58	3A	072	€#58;	:	90	5A	132	€#90;	Z	122	80	172	€#122;	z
27	1B	033	ESC (escape)	59	3B	073	€#59;	;	91	5B	133	€#91;	[123	81	173	€#123;	{
28	1C	034	FS (file separator)	60	3C	074	€#60;	<	92	5C	134	€#92;	\	124	82	174	€#124;	
29	1D	035	GS (group separator)	61	3D	075	€#61;	=	93	5D	135	€#93;]	125	83	175	€#125;	}
30	1E	036	RS (record separator)	62	3E	076	€#62;	>	94	5E	136	€#94;	^	126	84	176	€#126;	~
31	1F	037	US (unit separator)	63	3F	077	€#63;	?	95	5F	137	€#95;	_	127	85	177	€#127;	DEL

Source: www.LookupTables.com

Tabella ASCII Esteso

128	Ç	144	É	160	á	176	☒	192	Ł	208	ł	224	α	240	≡
129	ù	145	è	161	í	177	☓	193	ł	209	ŧ	225	β	241	±
130	é	146	Æ	162	ó	178	■	194	ŧ	210	ŧ	226	Γ	242	≥
131	â	147	ô	163	ì	179		195	ŧ	211	ł	227	π	243	≤
132	á	148	ò	164	ñ	180	†	196	—	212	ł	228	Σ	244	∫
133	ä	149	õ	165	Ñ	181	‡	197	†	213	ŧ	229	σ	245	∫
134	å	150	û	166	ª	182	‡	198	‡	214	ŧ	230	μ	246	÷
135	ç	151	ü	167	º	183	ŧ	199	‡	215	ŧ	231	τ	247	∞
136	è	152	ÿ	168	¿	184	ŧ	200	ł	216	‡	232	ϕ	248	°
137	é	153	ÿ	169	ƒ	185	‡	201	ŧ	217	ŧ	233	⊖	249	·
138	ê	154	Û	170	—	186	‡	202	ł	218	ŧ	234	Ω	250	·
139	í	155	ó	171	¼	187	ŧ	203	ŧ	219	■	235	δ	251	√
140	î	156	ê	172	½	188	‡	204	‡	220	■	236	∞	252	∞
141	ï	157	ÿ	173	¾	189	‡	205	—	221	■	237	ϕ	253	²
142	À	158	ŧ	174	«	190	‡	206	‡	222	■	238	ε	254	■
143	Á	159	f	175	»	191	ŧ	207	ł	223	■	239	∩	255	∩

Source: www.LoopupTables.com

Unicode

- Codifica di tutti i caratteri possibili
 - delle lingue attualmente vive
 - di alcune lingue morte
- Più di 110mila caratteri