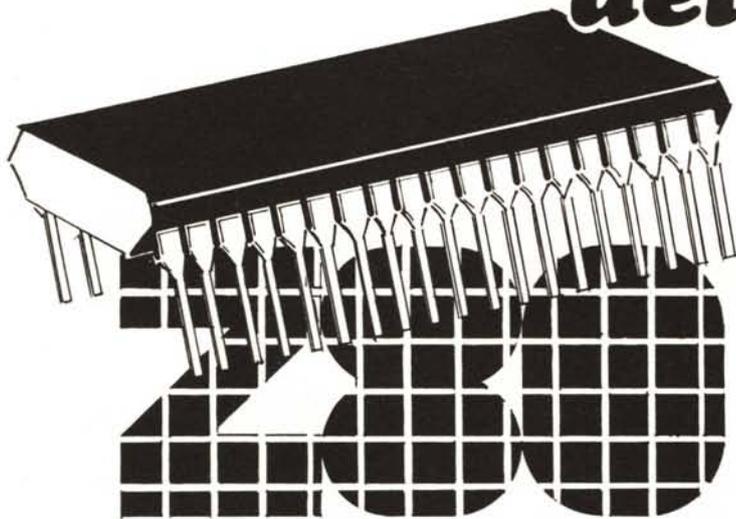


L'ASSEMBLER dello



di Pierluigi Panunzi

La gestione dello Z80

In questa puntata ci occuperemo delle istruzioni riguardanti la chiamata di subroutine ed il successivo ritorno al programma chiamante.

Prima di vedere quali istruzioni abbiamo a disposizione per la gestione delle subroutine, soffermiamoci un istante sul significato e l'importanza delle subroutine in un programma: evidentemente ci rivolgiamo a coloro che si sono avvicinati da poco al mondo della programmazione in assembler.

Supponiamo che il nostro programma, o meglio il nostro problema, richieda in parecchie parti di eseguire un certo gruppo di istruzioni fisse, indipendentemente dal punto in cui devono essere eseguite.

In questo caso è comodo e conveniente considerare questo gruppo di istruzioni come un mini-programma o meglio un "sotto-programma" a sé stante, avente cioè uno o più punti di ingresso ed uno o più punti di termine dell'elaborazione: è inutile dire che questi mini-programmi sono proprio le subroutine.

Perciò tutte le volte che desideriamo che, ad un certo punto del programma, venga eseguito un determinato gruppo di istruzioni (la nostra subroutine), allora dovremo semplicemente "chiamare" tale subroutine, con un'apposita istruzione, per l'appunto la CALL, che in italiano significa proprio "chiamare".

Vediamo dunque come si comporta in questi casi il computer, dal punto di vista esecutivo delle varie istruzioni.

Supponiamo dunque di aver eseguito una certa istruzione: subito dopo troviamo

una "CALL" ad una certa subroutine.

In questo caso il calcolatore deve saltare alla subroutine per eseguirne le istruzioni, ma saltando deve "ricordarsi" del punto del programma in cui si trovava all'atto della chiamata stessa: questo perché una volta eseguite le istruzioni della subroutine, *dobbiamo* ritornare indietro al programma "chiamante".

Se qualcuno non è convinto di questo fatto di dover per forza ritornare al programma vero e proprio, gli rispondiamo domandandogli perché allora invece di usare una CALL non ha usato una semplice istruzione di "jump" che non fa altro che saltare ad un'altra locazione di memoria...

Perciò alla fine della nostra subroutine *dobbiamo* porre un'istruzione di "RET" (ritorno) al programma.

Vediamo ora in dettaglio questo meccanismo di "salto con ritorno".

Dicevamo che avevamo eseguito una certa istruzione dopodiché abbiamo incontrato una CALL: a questo punto per essere sicuri di ritornare proprio in quel punto, o meglio all'istruzione successiva alla CALL, al termine dell'esecuzione della subroutine il calcolatore deve memorizzare da qualche parte l'indirizzo di memoria a cui si trova l'istruzione "successiva" alla CALL.

Attenzione al fatto che l'indirizzo deve essere quello dell'istruzione successiva e NON quello della CALL, altrimenti una volta eseguita la subroutine, il programma tornerebbe alla chiamata, rieseguirebbe la subroutine e così via all'infinito!

Per memorizzare appunto tale fatidico "indirizzo di ritorno", il processore usa

quella particolare struttura di celle di memoria chiamata "stack" (che tradotto suona come "pila").

Quando la subroutine è stata eseguita per intero, allora il processore si riprenderà dallo "stack" l'indirizzo di ritorno ed automaticamente proseguirà nell'esecuzione del programma originario.

Supponiamo ora di complicare ulteriormente il problema: il nostro programma ad un certo punto chiama una subroutine (che per comodità chiamiamo "A") e quest'ultima a sua volta ne chiama un'altra (che chiamiamo "B").

In questo caso è auspicabile (ed infatti è proprio così!) che il processore:

1) esegua il programma fino alla chiamata di "A"

2) esegua le istruzioni della subroutine "A" fino alla chiamata della seconda subroutine ("B")

3) esegua per intero la seconda subroutine "B"

4) ritorni correttamente al punto in cui aveva sospeso l'esecuzione della prima subroutine "A"

5) termini di eseguire la subroutine chiamata per prima ("A")

6) ritorni nell'esatto punto in cui il programma principale era stato "abbandonato" per eseguire la prima subroutine

7) continui l'esecuzione del programma.

Questa volta il processore dovrà corrispondentemente:

1) salvare nello "stack" l'indirizzo di ritorno "al" programma principale

2) salvare nello "stack" l'indirizzo di ritorno "alla" subroutine "A"

3) —

4) riprendere dallo stack l'ultimo indirizzo ivi riposto

5) —

6) riprendere dallo stack l'altro indirizzo rimasto.

7) —

Attenzione al fatto che l'indirizzo posto nello stack nel punto 1) deve essere ripreso nel punto 6), mentre quello depositato nello stack al punto 2) deve essere "ripescato" al punto 4), e *non viceversa*...

Per comprendere meglio il significato di questo apparentemente complesso meccanismo, sfruttiamo il solito paragone: quello della "pila di piatti".

In questo modo di vedere, il primo indirizzo salvato è rappresentato da un piatto posto per esempio su di un tavolo. Ora il secondo indirizzo rappresenta un secondo piatto che ora "impiliamo" sopra al primo.

Quando dobbiamo "ritornare indietro" dobbiamo riprendere i nostri piatti nel modo più naturale e cioè prenderemo per primo quello che sta "sopra" (e cioè l'ultimo che abbiamo messo) e solo dopo quello che sta "sotto".

L'esempio diventa ancora più colorito se supponiamo di riscontrare il caso in cui la seconda subroutine ne chiama una terza, questa ne chiama una quarta e così via, supponiamo fino ad una decima subroutine.

In questo caso la nostra pila di piatti sarà cresciuta parecchio, fino a 10 piatti, ed a meno che non vogliamo ottenere risultati catastrofici (che si riflettono ovviamente sul nostro programma), dobbiamo, al ritorno da una subroutine, prendere per forza il piatto posto sulla sommità della pila.

In termini tecnici questo meccanismo, che avviene automaticamente all'esecuzione di un'istruzione CALL, prende il nome di "L.I.F.O." dall'inglese "Last In First Out" e cioè "l'ultimo venuto ("in") è il primo ad uscire ("out")". Il fatto che poi qualcuno trovi uno strano rapporto tra la prima parola della sigla ed il concetto stesso di "pila di piatti", è solo dovuto al fatto che vede troppa pubblicità in televisione!

A parte gli scherzi, quando il processore nell'eseguire una subroutine incontra un'istruzione RET, allora estrae dallo stack l'ultimo indirizzo che vi aveva deposto e salta a tale locazione di memoria per riprendere quanto aveva lasciato in sospeso.

Prima di vedere in dettaglio le istruzioni di CALL e RET nello Z80, ricordiamo che le subroutine non sono prerogativa del linguaggio macchina, ma si incontrano anche nei linguaggi ad alto livello, dei quali facciamo due esempi:

— in Basic si identifica generalmente una subroutine con il numero di linea da cui comincia (ad esempio GOSUB 1000) mentre la subroutine stessa termina con un'istruzione RETURN

— in Pascal le subroutine prendono il nome di "procedure" (che può essere letto sia in inglese che in italiano): la chiamata avviene semplicemente indicando il "nome" della procedura (ad esempio FATTO-RIALE), mentre la procedura, che nell'esempio si prefigge lo scopo di calcolare il fattoriale, deve essere racchiusa tra una "begin" ed una "end".

Le istruzioni CALL

Analogamente alle istruzioni di salto che abbiamo conosciuto la scorsa puntata, anche in questo caso non abbiamo un'unica istruzione di chiamata a subroutine, ma abbiamo anche le "chiamate condizionate".

Perciò accanto all'istruzione CALL nnnn che effettua la chiamata alla subroutine posta all'indirizzo "nnnn", abbiamo le seguenti possibilità:

CALL NZ,nnnn
CALL Z,nnnn
CALL NC,nnnn
CALL C,nnnn
CALL PO,nnnn
CALL PE, nnnn
CALL P,nnnn
CALL M,nnnn

Le prime due sono chiamate a subroutine condizionate allo stato del flag di Zero, la terza e la quarta sono legate al flag di Carry, la quinta e la sesta dipendono dal

flag di Parity, mentre infine le ultime due sono condizionate dal flag di Sign.

Valgono in questo caso tutte le considerazioni fatte la scorsa puntata nei confronti delle istruzioni di "salto incondizionato e condizionato".

C'è da dire che in questo caso non esistono le chiamate "relative" e cioè l'indirizzo "nnnn" posto nella istruzione CALL sarà sempre e soltanto un indirizzo assoluto, effettivo di memoria.

Le istruzioni RET

Abbiamo in questo caso un'analogia completa con le istruzioni CALL.

Infatti a fianco alla RET che fa ritornare incondizionatamente al programma chiamante, esistono altre otto istruzioni di ritorno "condizionato":

RET NZ
RET Z
RET NC
RET C
RET PO
RET PE
RET P
RET M

che effettuano il ritorno soltanto se è verificata la condizione specificata.

Con questo abbiamo concluso l'argomento delle subroutine e diamo l'appuntamento ai lettori alla prossima puntata con altre istruzioni dello Z80. **MC**



COMPUTER SYSTEMS



apple® computer

Rivenditore e centro assistenza autorizzato

ROMA - Via G. Lanza 101-103-105 (tra Via Merulana e Via Cavour) Tel. 738224-738854

M fermata Vittorio Emanuele (linea A) - Via Cavour (linea B)

OSTIA LIDO - Via A. Carabelli 108-110-112

Tel. 5697686

(tra Via Isole Capoverde e Via delle Azorre)

GROTTAFERRATA - Via Trento 32-34 (Centro dimostrativo)

Sabato aperto anche pomeriggio

IL BITTEGONE

00123 ROMA VIA U. COMANDINI 49 TEL. 06/6133 025-79 20 559 TR.621166 FEPAG II

IMPORTATORI DIRETTI DALL' ESTREMO ORIENTE

di Felice Pagnani

SUSY 5
SISTEMI

E' IBM PC COMPATIBILE

ECCEZIONALE

IL SISTEMA PER CHI INIZIA

128K 1 DRIVE 360K, USCITA MONOCROMATICA GRAFICA 7.20X390	408.000
USCITA STAMPANTE E STAMPANTE GRAFICA, VIDEO 25MHZ E BASCULAMENTO, DOS 2.0 TASTIERA ALIM. 130W, 8 SLOTS DI ESPANSIONE	3.575.000
SUSY 5/HD Come sopra ma con Hard Disk 10 Mbytes, 6.000.000	6.000.000
SUSY5 BASE, 728K, grafica monocromatica, uscita parallelo monitor un floppy	2.990.000

COLLEGAMENTI LOCALI & REMOTI

DATANETS coppia di schede DATANET per rete locale con cavo e software	2.550.000
DATANETS scheda DATANET per rete locale di un sistema aggiuntivo	1.300.000
PCOK scheda di emulazione di terminali 3278/3279 tramite control unit 3274 e cavo coassiale	2.600.000
PC5251 scheda per il collegamento via modem con sistemi IBM 34, 36 o 38	1.400.000
IRMA scheda per l'emulazione di terminali 3278 tramite control unit 3274 e cavo coassiale	\$ 1.700
ACCOPLIATORE AGLISTICO	300.000
MODEM 300 BAUD	260.000

MONITORS & ACCESSORI

MONITOR MONOC. ALTA RES. 25MHZ BASCULANTE 18.4KHZ/H'	330.000
MONITOR COLORE RGBI	889.000
MONITOR COLORE BARCO	2.100.000
CABINET E ACCESSORI	277.800
TASTIERA SLIM	254.000
PCMOUSE ottico completo di interfaccia RS232, alimentatore e tavoletta	640.000
PCPAINT come sopra con l'aggiunta del SW PCPAINT	740.000
STAMPANTI:	
80CHR/120 CPS	740.000
80CHR/180 CPS	990.000
136CHR/180 CPS	1.420.000
mancherita	1.250.000
mancherita	720.000
mancherita	2.750.000

CADD PER IBM

Sistema grafico per IBM adatto per applicazioni in architettura, meccanica ed elettronica comprende:

MONITOR 0.31 dot pitch colore	1024X1024 pixels, NEC
VIDEO CONTROLLER 512 KB RAM	720800 nsec velocità di plotting
campo di indirizzamento	32.000 x 24.000 pixels. Il controller video è direttamente connesso al BUS del PC IBM mediante un cavo ed una scheda di adattamento.
TABELLA risoluzione:	0.125 mm, dim. 305x305 mm. RS232.
PLOTTER	40 cm/sec DIN A3/A4
Completo di SW	26.000.000

SUSY 2
SISTEMI

E' APPLE 2 COMPATIBILE

48K pad numerico	774.000
64K pad num. e t.funz.	832.000
64K tastiera separata, 2Mbytes su 2 minifl. tastiera separata e contenitore tipo IBM, monitor ergonomico 25MHz, DOS 3.3	3.300.000
Pascal 1.1 e CP/M 2.2, stampante grafica	3.300.000
UN SISTEMA PER CHI INIZIA	
SUSY 2 48K UN FLOPPY SLIM UN MONITOR 12" 25MHz	1.400.000
SCHEDE MADRI SUSY 2	
48K zpccli	381.000
64K	465.000

INTERFACCE PER SUSY 2

DISK DRIVE CARD	78.000
DISK DRIVE DOUBLE/FACE	121.800
PRINT INT. EPSON CARD	73.500
PARALLEL PRINTER CARD	72.200
UNIVERSAL PRINTER CARD	134.400
PRINT CABLE	41.800
LANGUAGE CARD	102.300
18K RAM CARD	102.300
INTEGER CARD	102.300
280 CP/M CARD	77.400
80X24 VIDEO CARD	129.500
80X24 VIDEO W/SWITCH	169.700
RS232 CARD	109.100
COMMUNICATION CARD	102.100
7710 ASYNCHRONOUS CARD	200.200
FORTH CARD	89.400
PARALL. CARD W/32KRAM	296.700
BUFFER CABLE 2/PCS	58.800
6522 CONTROL CARD	84.800

IEE488 CARD	225.800
SPEECH CARD W/SW	88.100
128K RAM SATURN W/SW	505.300
6809 CARD W/FLEX MANU.	358.800
MUSIC SYSTEM W/SW	128.900
WILD CARD	88.700
PAL CARD W/MODULATORE	110.000
AD/DA CARD W/SW	337.500
EPROM WRT (2716-32-64)	127.300
CLOCK CARD W/SW	124.400
OLIVETTI PRAXIS CARD	234.300
IBM CARD 8088 W/SW	625.200
RGB CARD W/CABLE	131.800
UNIVERSAL PARALL.PRINT	131.300
8748/49 MPU PROGRAMMER	429.800
JHON BELL A/D CARD	162.300
TELEFAX CARD	556.800
APPLI C. 280,64K W/SW	649.900
IC TEST CARD W/SW	255.900
INTER.DRIVE 2 MBYTES	385.000

ACCESSORI PER SUSY 2

RF MODULATOR	17.800
JOYSTICK	23.500
DESK TOP JOYSTICK	33.900
JOYSTICK AUTOCENTERING	47.100
JOYSTICK AUTO QUIK FIRE	55.400
91 FUNCTION EXT.APPLE	133.200
FAN	29.500
COOLING FAN W/CABLE	75.100
LIGHT PEN HI RES.	393.400
TAVOLETTA GRAFICA	123.200
DRIVE 5" SINGLE HEAD	400.800
DRIVE 5" DOUBLE HEAD	748.400
DRIVE 5" 64KBYTES	500.600
DOUB.DRIVE MULTITECH	1.042.900
TAST. SUSY 2 MULTITECH	235.700
TASTIERA CON PAD NUM.	190.900
MONITOR COLORE RGB 14"	560.000
MONITOR B/N,VERDE,OCRA	215.000
BASCULAMENTO EUROVIDEO	35.000

PRO-DOS (COMPATIBILE)

Il PRO-DOS compatibile con tutti i compatibili. Riconosce l'ambiente in cui si trova e ci si adatta automaticamente. Il dischetto: 35.000

Un altro modo per rendere compatibile con il PRO-DOS il proprio compatibilis EPROM PRO-DOS UNIVERSALE si inserisce al posto della ROM F8 e non è più necessario modificare i dischetti. 2KEPROM 38.500 4KEPROM 42.500

Tutti i sistemi SUSY 2 sia da 48 sia da 64K RAM vengono già forniti compatibili con la EPROM di PRO-DOS.

CABINETS PER SISTEMI
Monitor ergonomico con spazio per schede e alim. 100.000
Computer in due pezzi 75.000

PRINTER AD AGHI
STAMPANTI A MARGHERITA
SP100 100 CPS GRAFICA 642.000
SP120 120 CPS GRAFICA 730.000
MACCHINA DA SCRIVERE E STAMPANTE A MARGHERITA 700.000

ECCEZIONALE

XY PLOTTER

- rullo da 114 e 210 mm.
- Numerous intelligent funct.
- 4 Colour graphics.
- OHP Film Drawings.
- Usable as printer.
- Support graphics and special graphic symbols

760.000

MATERIALI DI CONSUMO

DISCHETTI 5" ACUTRACK	37.000
DATA LIFE SF/DD BOX	40.000
RHONE POULENK SF/DD	40.000
RHONE POULENK DF/DD	60.000
CARTA 2000 FOLII 80 C	40.000
CARTA 1000 FOLII 80 C	20.000
CONTENITORE DISCHI	44.000
dischetti da 8" sf/sd e df/dd	

IVA

Tutti i prezzi sono IVA esclusa, pagamento in contanti, spedizioni in tutta Italia contrassegno. GARANZIA 3 MESI.

COMUNICAZIONE

Nonostante i massimi storici che il dollaro conquista noi abbiamo aumentato del minimo. I prezzi che pagate sono quelli indicati, quindi niente sorprese all'atto della fatturazione per il mese di MARZO.

DLR.

LA PROPOSTA DEL MESE

CONTINUA CON ENORME SUCCESSO LA PROPOSTA DEL MESE CHE PARTITA A SETTEMBRE NON ACCENNA AD ESAURIRSI SI TRATTA DI Una scatola con 10 dischetti con tutti i migliori giochi del mondo piu' un joystick analogico autocentering. In tutto 10 megabytes di tutto di divertimento 90.000 Solo la scatola 60.000

SUSY SUPER GRAPHIC

nostra esclusiva Progettazione e

SUSY SUPER GRAPHIC
Trasforma un SUSY (o compatibile) un APPLE II E in un potentissimo sistema grafico. I piu' alti livelli della grafica per impieghi professionali prima irraggiungibili per l'alto costo ora sono alla portata del piu'.

Risoluzione 1Mega pixels (1024 x 1024 b/n o 512x512 4 piani colore). Generazione di disegni da hardware vettoriali, cerchi e rettangoli, possibilità di PAN, SCROLL, ZOOM (fino a 16 volte). Uscita

RGB. 128 KRAM a bordo, processore NEC7220 (16bit). Software fornito; interprete e PAINT consente l'uso di una tavoletta grafica digitale o del joystick. 1.750.000

Schedino PIG-BACK SSG per avere un uscita RGB lineare, videocomposito e una tavoletta di 4096 colori 275.000

Software opzionale: PRIMITIVE consente l'uscita dal PAINT su basic con possibilità di aggancio di set di caratteri e figure, generare delle funzioni. 175.000

RAM DISK consente di usare la scheda anche come disco virtuale sotto DOS 3.3 e sotto CP/M 100.000

SCHEDE IN STD-BUS 280

ADATTE PER CONTROLLI INDUSTRIALI IN AMBIENTI AD ELEVATO STRESS - FUNZIONAMENTO 24 ORE

SU 24 - ELEVATISSIMA AFFIDABILITA' - ADATTE ANCHE PER GESTIONALI CON IMPIEGO GRAVOSO.

CPU-I/O 64KRAM 2 seriali 1 parallela zocc.EPROM 750.000

CPUI-I/O cs.senza RAM 390.000

FC2 Floppy Contr.OD 515.000

DR1 RAM 64K 470.000

DR2 RAM 256K 880.000

SPP1 4 p.seriali 407.000

Bn1 8 zoccoli Byte W. 242.000

AD1 4 p.parallele 319.000

accessori: 96.000

AD1 adapter Winchest. 96.000

BOX 4 posti scheda 180.000

BOX 8 con terminazioni 220.000

BOX 8 con terminazioni 270.000

PAL/20 PAL PROGRAMMER 900.000

PG/128 EPROM PROG 700.000

VB8 033 scheda video 80X24 RS232 320.000

SOFTWARE DI SUPPORTO ALLE SCHEDE: ADATTAMENTO CP/M, MP/M MULTISERS, BASIC RESIDENTE.

UN TERMINALE IN UFFICIO UNO A CASA E I DATI.....IN TASCA Un nuovo concetto di portatilità:

SISTEMA 10 POCKET 280 4MHZ, 2 porte seriali, 1 parallela, 64KRAM, 10Mbytes Winchester, 700Kbytes minifloppy. CP/M. Piu' piccolo di un beauty-case a lire 4.500.000

produzione

GESTIONALE FP10M MODULARE ESPANDIBILE

10Mbytes + 1 Floppy 5", 256K RAM, 2 PORTE SERIALI, 1 PARALLELA, 1 TERMINALE VIDEO, PREZZO 6.200.000

OGNI TERMINALE IN PIU' 900.000

GESTIONALE FP10M SBC, 10MBYTES + 700 KBYTES minifloppy. 64K RAM. Interfacce 2 seriali, 1 parallela. 1 video 5.500.000