Z80 uC esecuzione di programmi

Per immettere un programma e farlo funzionare si deve eseguire la seguente sequenza.

1)Prendere il controllo del bus mediante l'interruttore **BUSREQUEST**; si ha il controllo quando é acceso il led verde **BUSAK**;

2)Mediante i pulsanti **UP o DOWN** posizionare il contatore degli indirizzi (ADDRESS COUNTER) nella locazione di memoria da cui si desidera far partire il programma; l'indirizzo relativo appare sul visualizzatore degli ADDRESS in forma esadecimale (nibble High e Low).

3)Impostare (in forma binaria) i byte delle istruzioni del programma da eseguire mediante gli interruttori 0-3, 4-7 (negli esempi seguenti tali byte vengono indicati come "dato");

4)Trasferire nella locazione di memoria indirizzata il dato impostato mediante il pulsante DATA WRITE;

5)Incrementare di uno la posizione dell'ADDRESS COUNTER mediante il pulsante UP;

6)Ripetere la sequenza 3-4-5 fino al termine del programma;

7)Per controllare l'esattezza dei dati impostati si può decrementare l'ADDRESS COUNTER mediante il pulsante DOWN verificando, locazione per locazione, il contenuto della memoria e correggendo gli eventuali errori:

8)Restituire i bus alla CPU mediante l'interruttore BUSREQUEST (il led verde si spegne);

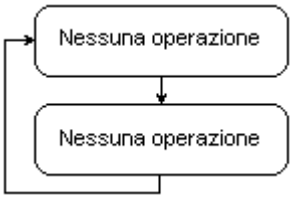
9)Premere momentaneamente il pulsante di RESET; si accende il led rosso di RUN e la CPU cerca la prima istruzione da eseguire in 0000_{HEX}.

Z80 Prima esercitazione

Creazione di un loop:

Questo programma fa funzionare il processore in un loop senza fine consentendo di osservare dei segnali periodici e di verificare le caratteristiche.

Diagramma di flusso:



indirizzo	dati	label	istruzione	commento
00	00	INIZIO:	NOP ;	"nessuna operazione
01	00		NOP ;	come sopra
02	C3 00 00		JP INIZIO ;	salto indietro alla prima istruzione

Dopo aver memorizzato il programma farlo eseguire con il clock interno ad 1 MHz. Sincronizzare esternamente l'oscilloscopio a doppia traccia con il segnale M1*; visualizzare su una traccia il clock della CPU (pin 6) e sull'altra traccia i seguenti segnali:

- MI* (pin 27) ciclo di fetch del codice operativo
- MREQ* (pin 19) richiesta di accesso in memoria
- RD* (pin 21) lettura della memoria
- D0 (pin 14) dato meno significativo
- A0 (pin 30) indirizzo meno significativo
- RFSH* (pin 26) segnale di refresh

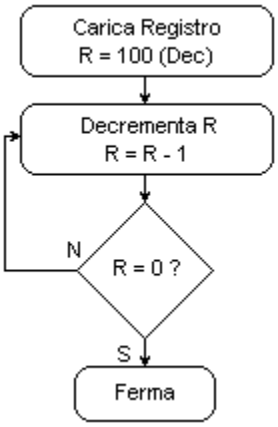
Disegnare l'andamento, la fase ed i tempi rispetto al clock, delle forme d'onda osservate e verificarne l'accordo con i diagrammi temporali riportati nella descrizione generale dello Z80.

Z80 Prima esercitazione

Programma di temporizzazione

Questo programma é frequentemente usato per effettuare operazioni ad intervalli uniformi di tempo; per il conteggio viene usato il registro B.

Diagramma di flusso



indirizzo	Dati	label	istruzione	commento
00	06 64		LD B,100	; carica il registro B con 100 (decimale)
02	05	LOOP:	DEC B	; decrementa B : B=B-1
03	C2 02 00		JP NZ,LOOP	; salto indietro se il risultato non é zero,
06	76		HALT	; altrimenti ferma

Dopo aver memorizzato il programma, farlo eseguire con il clock interno ad 1 MHz; al termine dell'esecuzione il led rosso di RUN collegato al pin 18 (HALT*) della CPU si spegne.

L'istruzione DEC impiega 4 cicli T, mentre JP ne impiega 10; tutto il loop dura quindi 100 (contenuto di B) x (10 (istr. JP) + 4 (istr. DEC) x T dove T é il periodo del clock. Verificare la durata di questo loop, per varie frequenze del clock esterno (per es. 1 KHz, 10 KHz, 100 KHz); a questo scopo si inserisca sulla traccia 1 dell'oscilloscopio il segnale di RESET* e sulla traccia 2 il segnale di HALT*; come segnale sincronizzante si utilizzi quello della traccia 1. Per una velocità di scansione opportuna, agendo sul comando di livello del sincronismo, ogni volta che il pulsante di RESET viene rilasciato si potrà osservare sulla traccia 2 un segnale che indica la durata del programma. Poiché il massimo contenuto del registro B é 255 (FF_{HEX}), vi é un limite al ritardo che si può ottenere da questo programma (si consideri il clock di sistema di 1 MHz, che corrisponde ad un periodo di 1 ns; il massimo ritardo ottenibile é di 3584 ms). Si possono ottenere durate variabili a piacere aumentando il numero dei registri da decrementare; questo comporta naturalmente un aumento dello spazio di memoria utilizzato per effettuare il ritardo.

Si scriva un programma che effettua un ritardo di 10 sec alla frequenza di 1 MHz.

Z80 Prima esercitazione

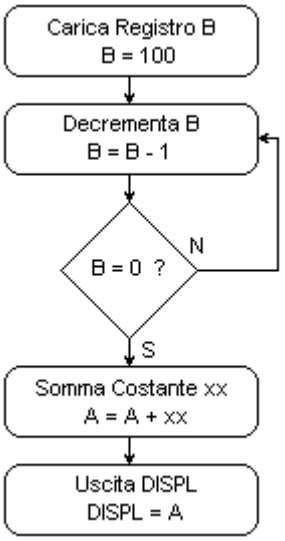
Programma di ingresso/uscita

I programmi che seguono mostrano la base delle tecniche di I/O dello Z80.
Quando viene eseguita una istruzione di uscita (o ingresso), che in assembler é scritta OUT (n),A (oppure IN A,(n)) dove n=operando (numero esadecimale) e A=Accumulatore, l'operando viene posto sulle linee di indirizzo A0-A7; poiché tali linee sono connesse ad un decodificatore, l'uscita del decoder corrispondente all'operando fornito in ingresso verrà resa attiva. Per esempio, l'istruzione OUT $03,A significa che viene resa attiva l'uscita Q3 del decoder.

Le periferiche utilizzate, a cui viene assegnato un nome simbolico, sono i visualizzatori dei dati (uscita) e gli interruttori di predisposizione dei dati (ingresso). Per connettere hardware tali periferiche é necessario (si veda lo schema elettrico);

- inserire un ponticello tra l'uscita desiderata (Q0-Q3) del decoder connessa al connettore I/O SELECT e il piedino del connettore KEYBOARD EN. ; viene quindi abilitata la periferica di ingresso;
- inserire un ponticello tra l'uscita desiderata (Q0-Q3) del decoder connessa al medesimo connettore e il piedino del connettore DISPLAY EN. ; viene quindi abilitata la periferica di uscita.

Z80 Prima esercitazione

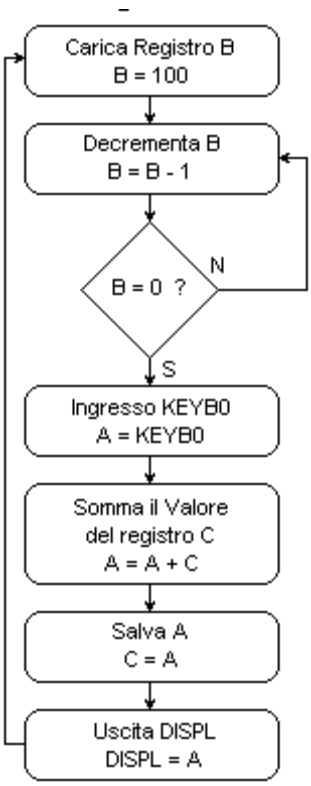


indirizzo	Dati	Label	istruzione	commento
00	06 64	INIZIO	B,100	; carica il registro B con 100
02	05	LOOP	DEC B	; decrementa B:B=B-1
03	C2 02 00		JP NZ,LOOP	; torna indietro se il risultato non é zero
06	C6 xx		ADD A,xx	; altrimenti somma la costante xx al contenuto dell'accumulatore
08	D3 01		OUT (O1),a	; metti il contenuto dell'accumulatore in uscita (DISPL)
0A	C3 00 00		JP INIZIO	; salta indietro e continua il ciclo

Nel memorizzare il programma scegliere il valore della costante xx da inserire nella locazione 07_{HEX};

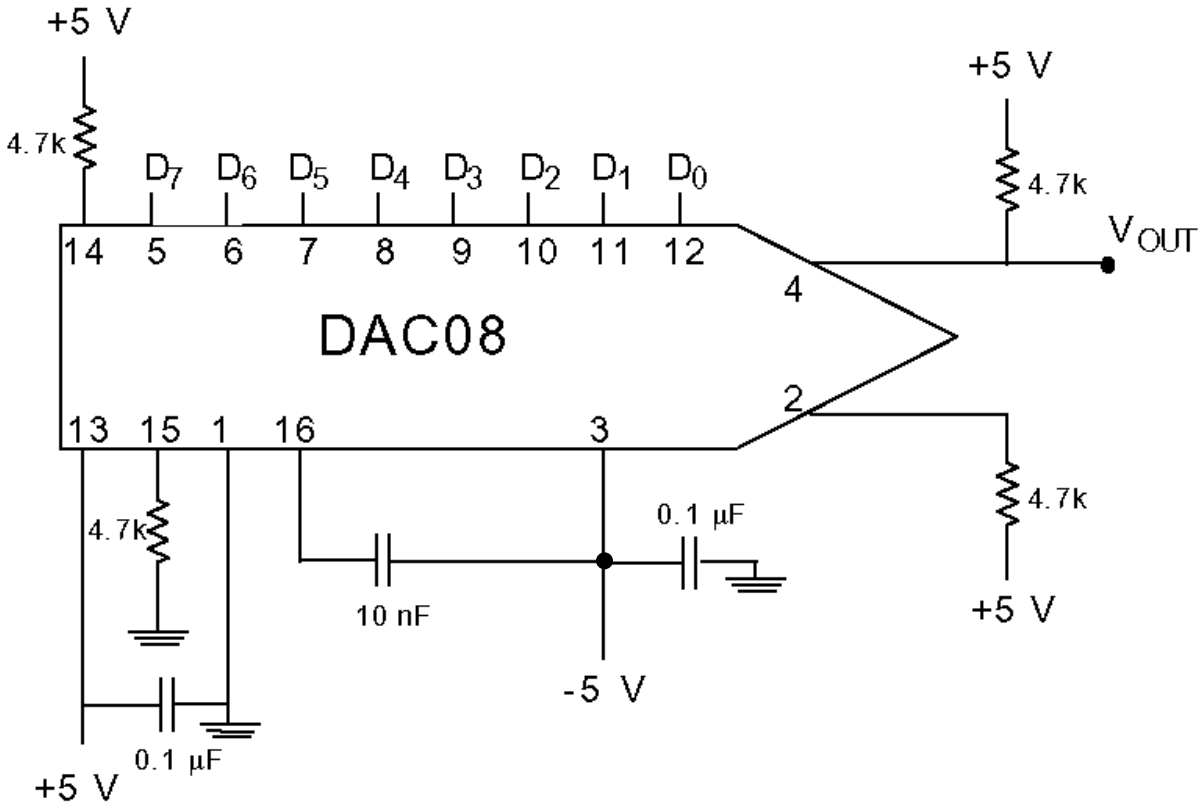
inserire quindi un clock esterno di 1 KHz e verificare, durante l'esecuzione, i valori presenti sul visualizzatore dei dati inserendo il valore 03 nella locazione 01 e con un clock esterno di 2-3 Hz osservare, sul visualizzatore degli indirizzi, il flusso del programma.

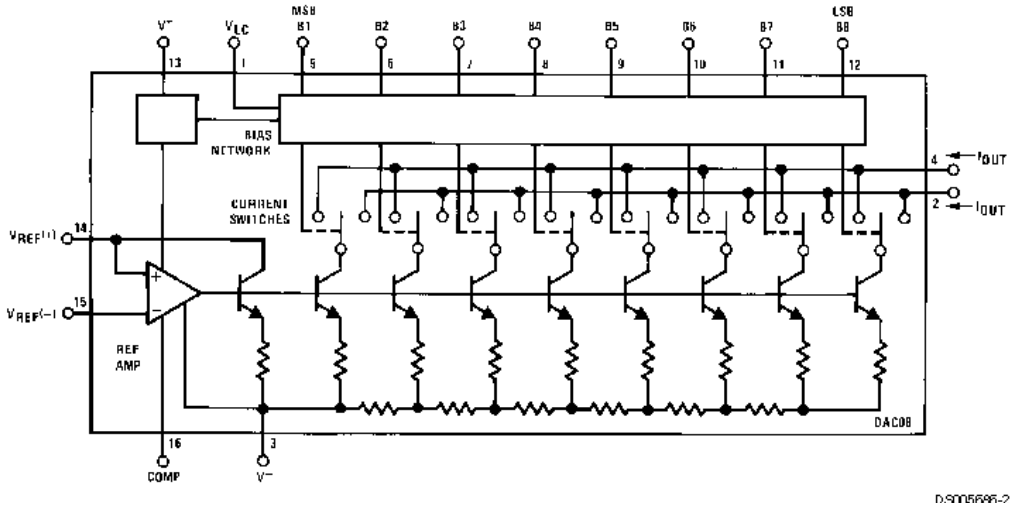
Z80 Prima esercitazione



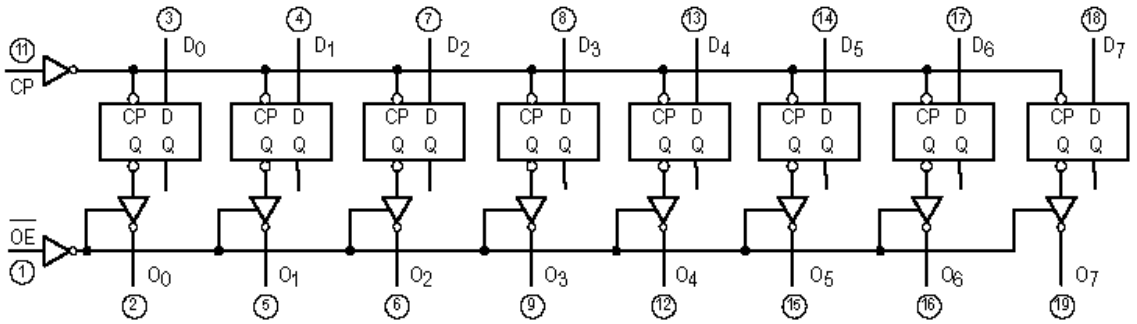
indirizzo	dati	label	istruzione	commento
00	06 64	INIZIO :	LD B,100	; carica il registro B con 100
02	05	LOOP :	DEC B	; decrementa B:B=B-1
03	C2 02 00		JP NZ,LOOP	; torna indietro se il risultato non é zero
06	DB 00		IN A,(OO)	; leggi gli interruttori e metti il valore hex nell'accumulatore
08	81		ADD A,C	; aggiungi ad A il totale contenuto nel registro C
09	00		NOP	; nessuna operazione
0A	4F		LD C,A	; trasferisci il nuovo totale nel registro C
0B	D3 01		OUT (O1),A	; metti in uscita il nuovo totale
0D	C3 00 00		JP INIZIO	; torna indietro e ripeti il ciclo

Z80 Seconda esercitazione

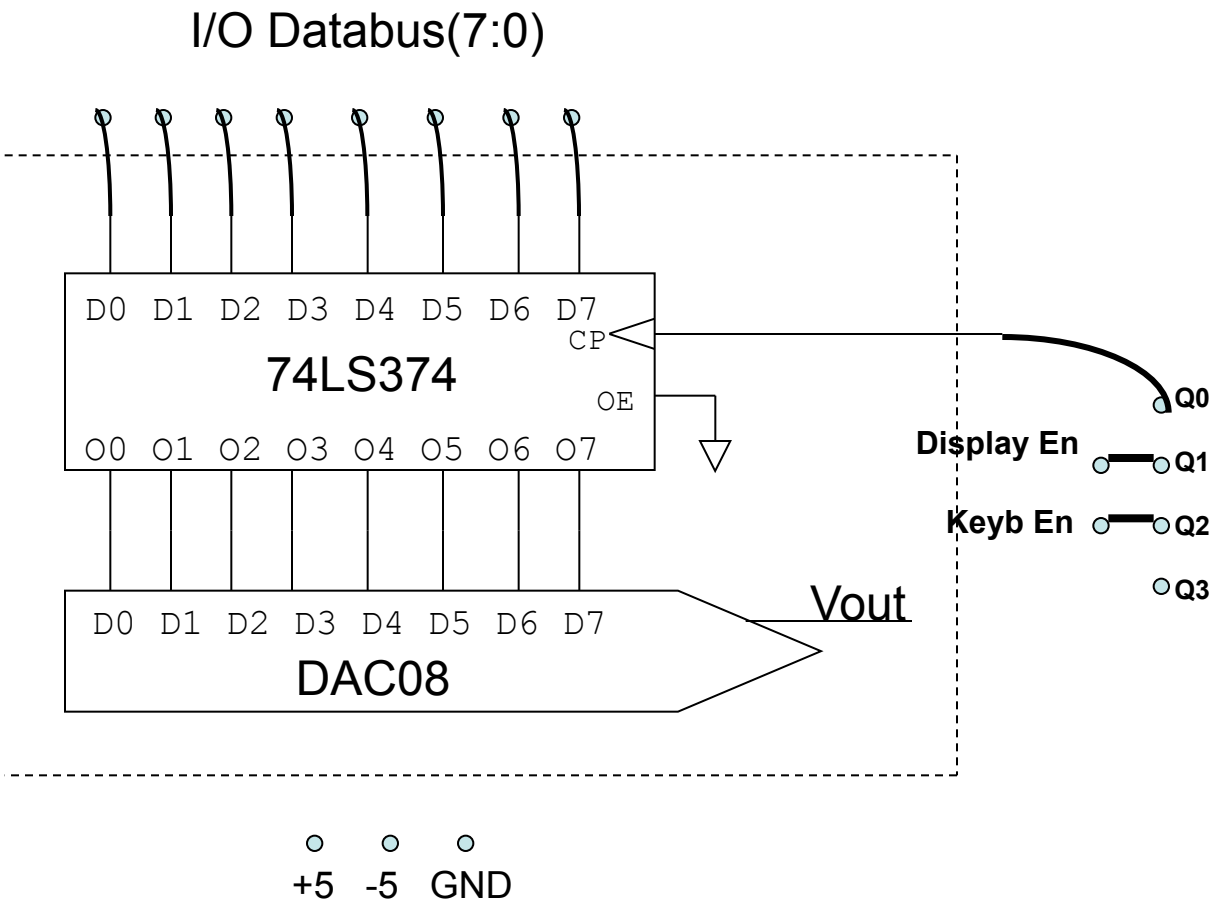




SN54LS/74LS374



Z80 Seconda esercitazione



1) Programma di calibrazione

indirizzo	Dati	label	istruzione	commento
00	DB 02	START	IN A,(02)	leggi gli interruttori e metti il valore hex nell'accumulatore
02	D3 00		OUT (00),A	scrivi dato su DAC
04	D3 01		OUT (01),A	scrivi dato su display
06	C3 00 00		JP START	

Z80 Seconda esercitazione

2) Generazione di una rampa positiva

indirizzo	Dati	label	istruzione	commento
00	DB 02	INIT	IN A,(02)	leggi gli interruttori e metti il valore hex nell'accumulatore (inizializza A)
02	3C	START	INC A	A = A+1
03	D3 00		OUT (00),A	scrivi dato su DAC
05	D3 01		OUT (01),A	scrivi dato su display
07	C3 02 00		JP START	

3) Generazione di una rampa negativa (DEC A)

4) Generazione di un “segnale triangolare”

indirizzo	Dati	label	istruzione	commento
00	DB 02	INIT	IN A,(02)	leggi gli interruttori e metti il valore hex nell'accumulatore (inizializza A)
02	3C	UP	INC A	A = A+1
03	D3 00		OUT (00),A	scrivi dato su DAC
05	FE FF		CP 255	Confronta A con 255 (1 se uguale)
07	CA 02 00		JPZ UP	Ripete se A =/ 255
0A	3D	DOWN	DEC A	A = A – 1
0B	D3 00		OUT (00),A	scrivi dato su DAC
0D	C2 0A 00		JPNZ DOWN	Ripete se A =/ 0
10	C3 02 00		JP UP	Ripeti l’intero ciclo up/down

5) Generazione di un “segnale triangolare” (periodo diverso)

Z80 Seconda esercitazione

6) Generazione di una forma d'onda qualsiasi

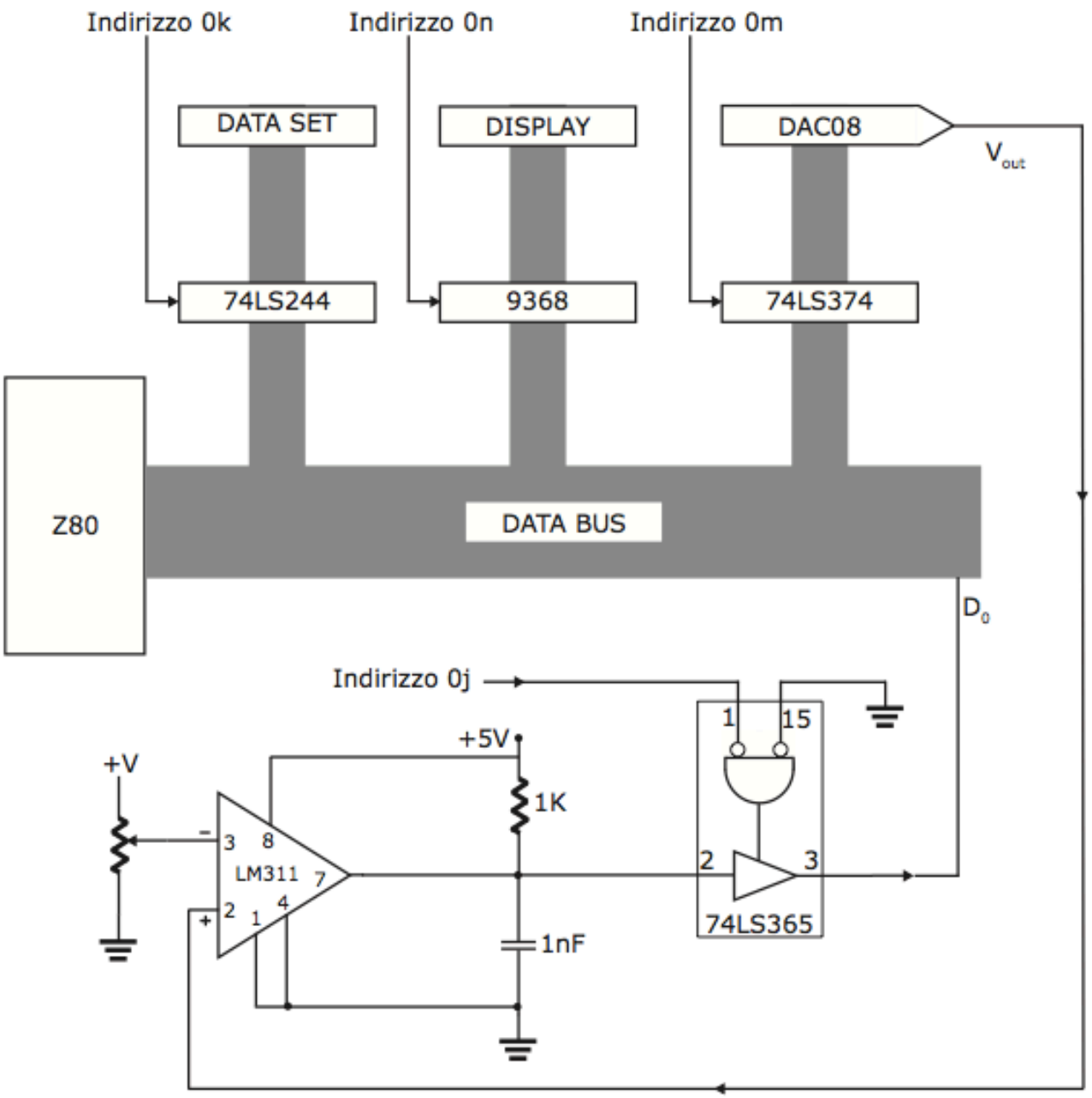
E' possibile costruire una forma d'onda a partire da una tabella contenuta nella memoria della scheda. Inserire in memoria jj dati (che rappresentano per punti discreti la forma d'onda desiderata) nelle locazioni da kk a kk + jj -1, e scrivere poi il seguente programma:

Indirizzo	Istruzione	Label	Mnemonico	Commento
00	06	start	LDB, jj	Carica in B il numero di dati
01	jj			
02	21		LDHL, 00kk	Carica in HL il primo indirizzo dati
03	kk			
04	00			
05	7E	loop	LDA,(HL)	Copia (HL) in A
06	D3		OUT(0n),A	Scrive sul DAC
07	0n			
08	23		INC HL	Incrementa HL
09	05		DEC B	B = B -1
0A	C2		JPNZ loop	Ripete per il nuoo dato
0B	05			
0C	00			
0D	C3		JP start	Ripete tutto il ciclo
0E	00			
0F	00			

Z80 Terza esercitazione

In questa esercitazione costruiremo un ADC.

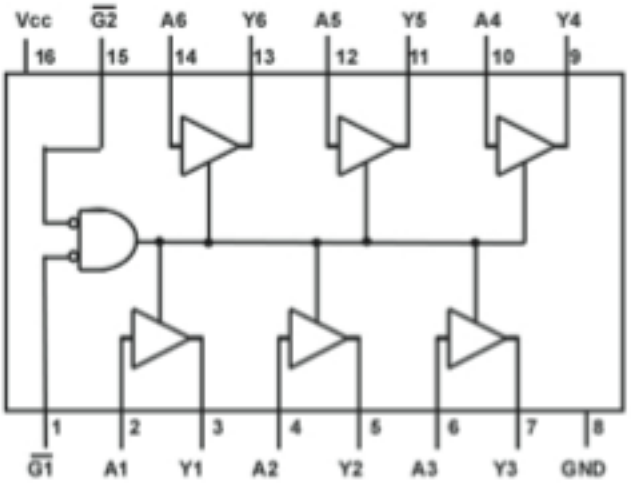
- Partendo dal circuito dell'esperienza precedente aggiungiamo un comparatore (LM311) all'uscita del DAC per confrontare con una tensione incognita V_x generata da un trimmer
- Nota: come prima cosa, montate e verificate il funzionamento del comparatore



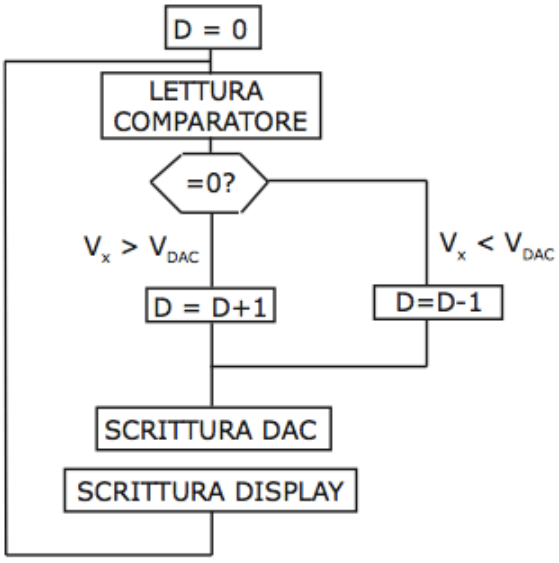
Z80 Terza esercitazione

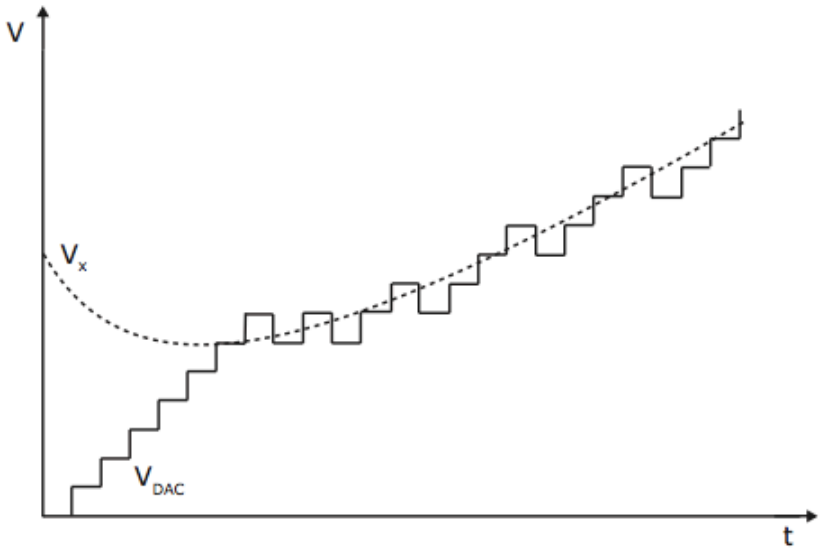
74LS365: buffer tristate abilitato da AND.
Permette la scrittura del valore del comparatore nel registro interno dello Z80

Input			Output
$\overline{G1}$	$\overline{G2}$	A	Y
H	X	X	Hi Z
X	H	X	Hi Z
L	L	H	H
L	L	L	L



Schema a blocchi del programma e comportamento aspettato dell'uscita del DAC (V_{DAC} insegue le variazioni di V_x





Z80 Terza esercitazione

Programma ADC tracking

Indirizzo	Istruzione	Label	Mnemonico	Commento
00	16		LD, 00	0 -> D
01	00			
02	7A	loop	LD A,D	D -> A
03	D3		OUT(0n), A	A -> Display
04	0n			
05	D3		OUT(0m), A	A -> DAC
06	0m			
07	DB		IN A,(0j)	Comparatore -> A
08	0j			
09	E6		AND 01	Maschera A con 00000001
0A	01			
0B	CA		JPZ up	se A=0 Vx > Vdac
0C	12			
0D	00			
0E	14	down	INC D	D = D-1
0F	C3		JP loop	RIPETI
10	02			
11	00			
12	15	up	DEC D	D = D+1
13	C3		JP loop	RIPETI
14	02			
15	00			