



*Istruzioni di tra-  
sferimento di dati*

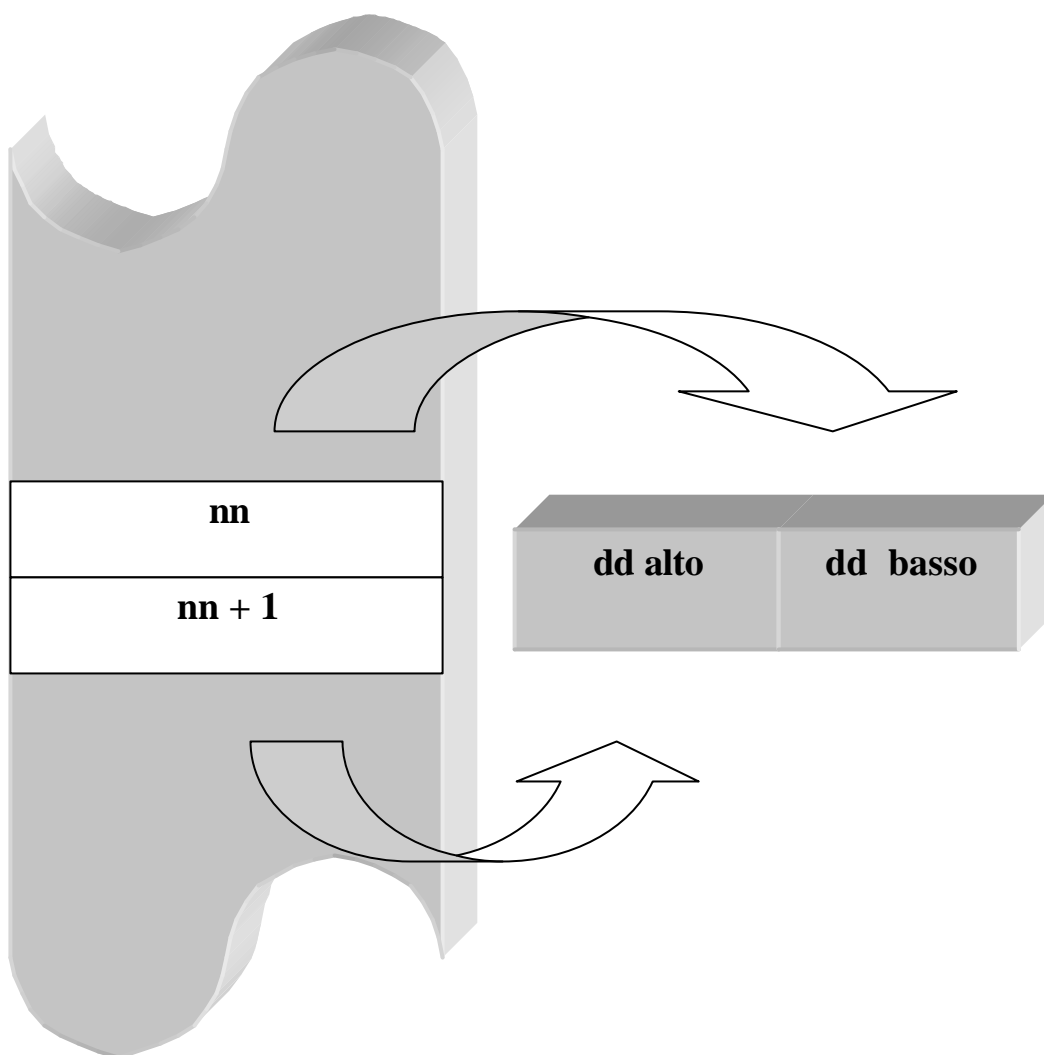
---

**LD dd, (nn)**

Con questa istruzione, il registro a 16 bit **dd** è caricato con il contenuto della coppia di locazioni di memoria di indirizzo **nn** ed **nn+1**.

La parte bassa del registro è caricata con il contenuto della locazione **nn** mentre la parte alta è caricata con il contenuto della locazione di indirizzo **nn+1**.

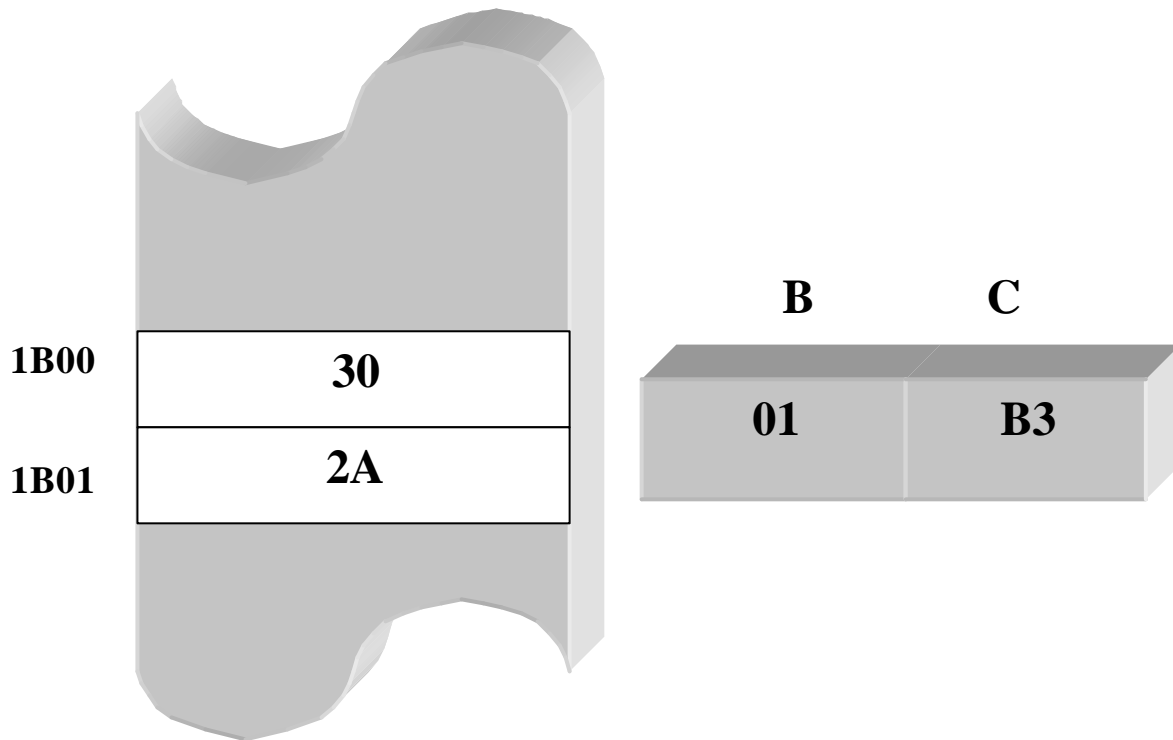
*L'istruzione non ha alcun effetto sul registro dei flag*



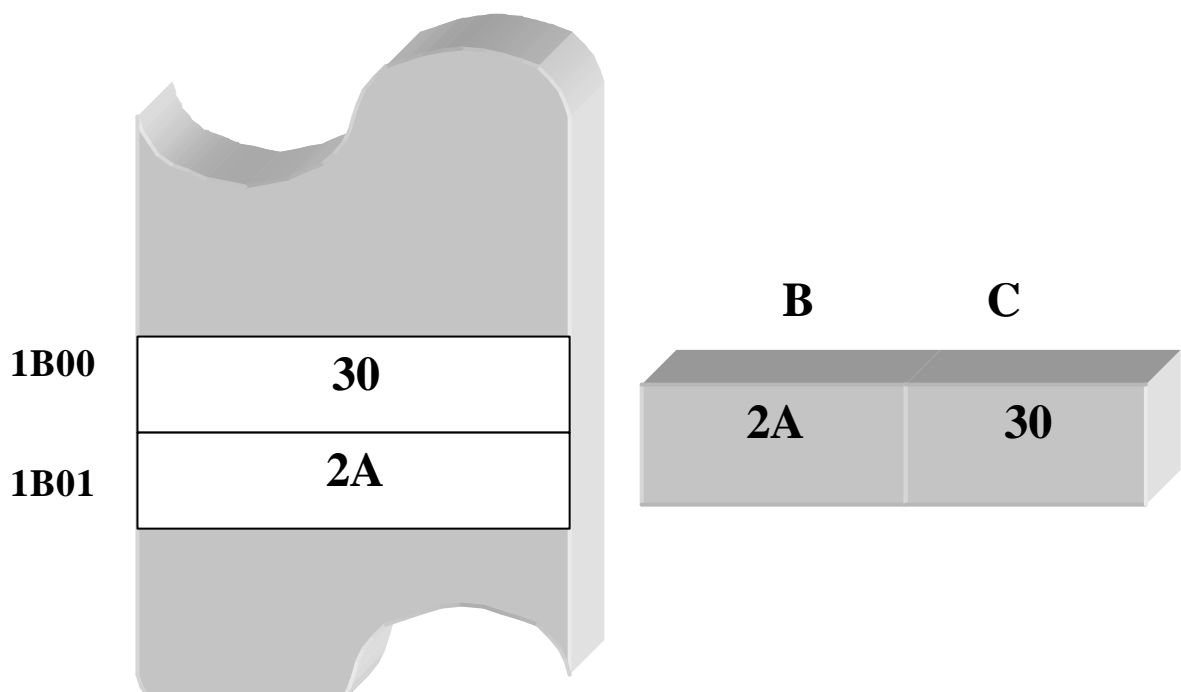
**Esempio:**

**LD BC, (1B00h)**

**Prima**



**Dopo**



IL registro dd può essere uno dei seguenti

BC

DE

HL

SP

IX

IY

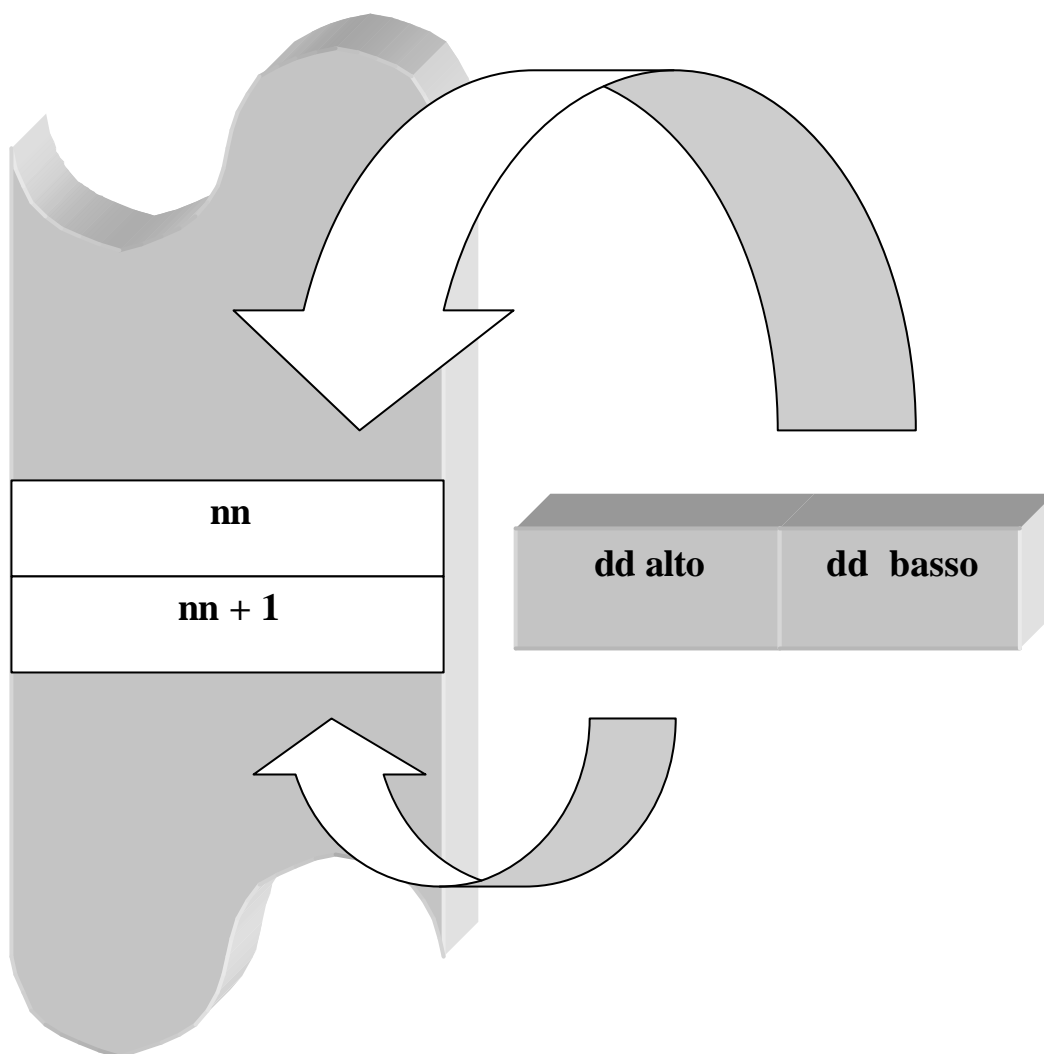
---

**LD (nn), dd**

Con questa istruzione, la coppia di locazioni di memoria di indirizzo nn ed nn+1 è caricato con il contenuto del registro a 16 bit dd.

La locazione di indirizzo nn è caricata con la parte bassa del registro dd e la locazione di indirizzo nn+1 è caricata con il contenuto della parte alta del registro.

*L'istruzione non ha alcun effetto sul registro dei flag*

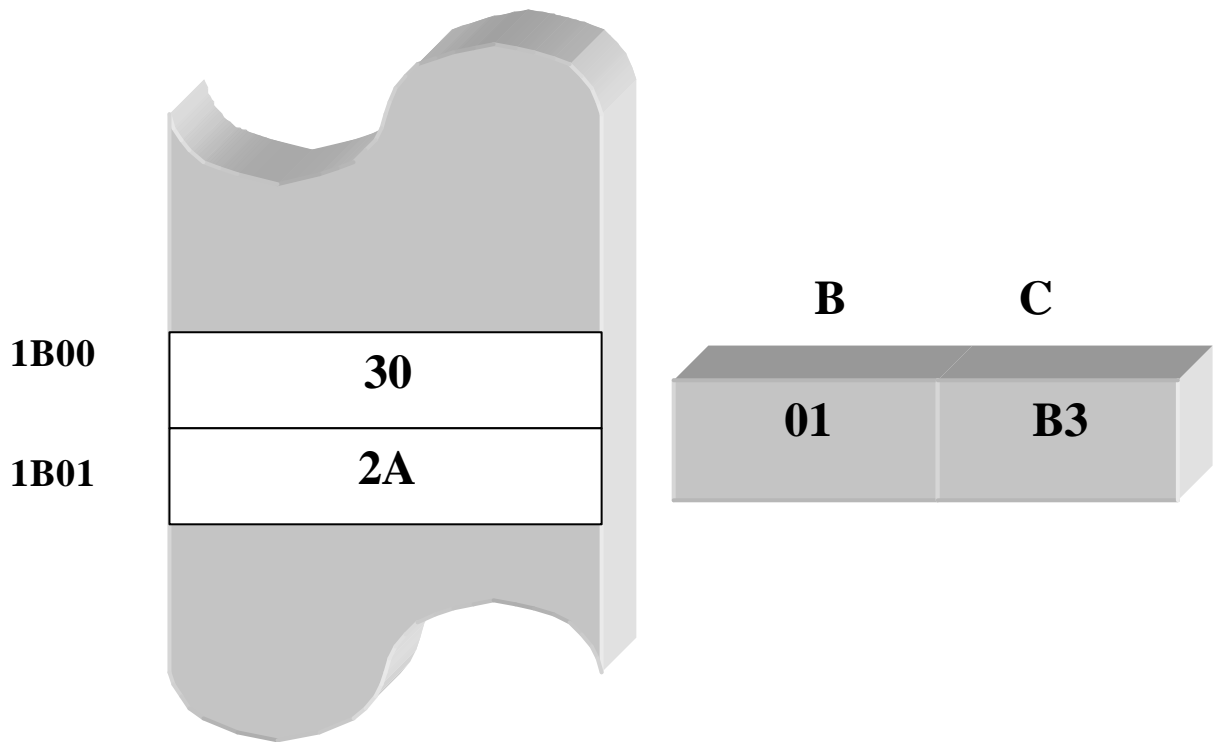


**Esempio:**

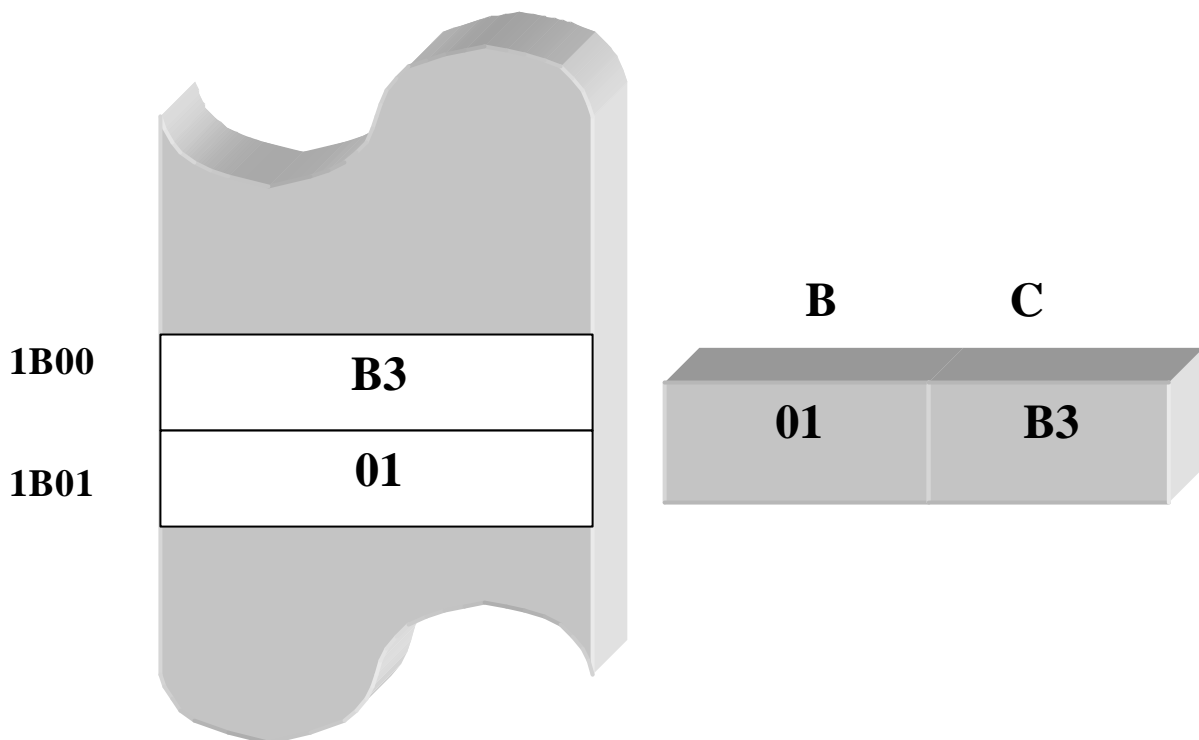
---

**LD (1B00h), BC**

**Prima**



**Dopo**



IL registro dd può essere uno dei seguenti

BC

DE

HL

SP

IX

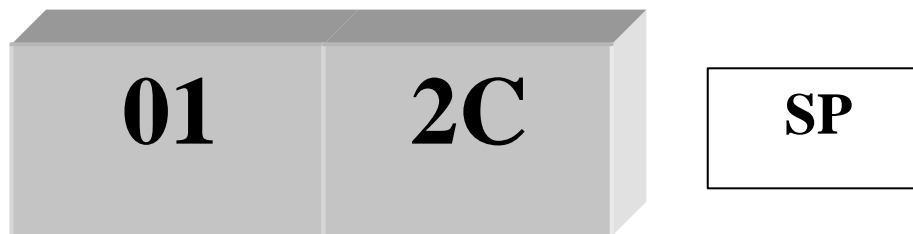
IY

---

**LD dd, nn**

La coppia di registri a 16 bit viene caricata con il dato immediato a 16 bit presente nell'istruzione.

*L'istruzione non ha alcun effetto sul registro dei flag*

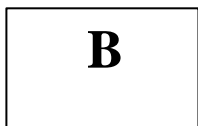
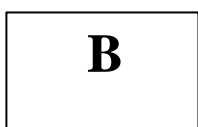
**Esempio:****LD SP, 3C20h****Prima****Dopo**

---

**LD r, n**

IL registro ad otto bit r viene caricato con il dato ad otto bit immediatamente espresso nell'istruzione.

*L'istruzione non ha alcun effetto sul registro dei flag.*

**Esempio****LD B, 23h****Prima****Dopo**

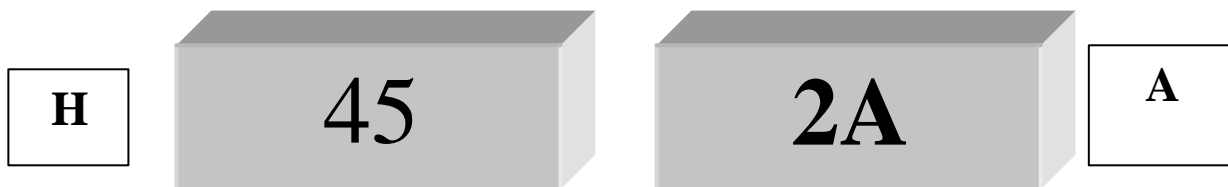
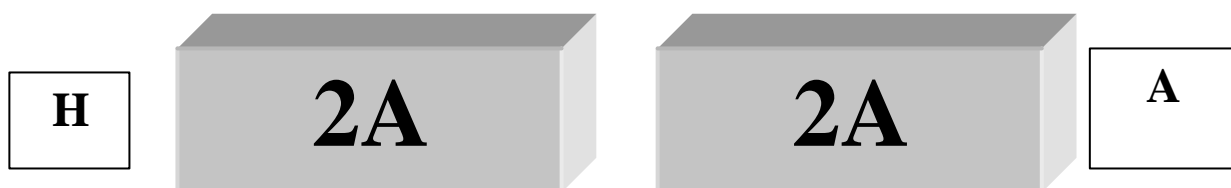
IL registro r può essere uno dei seguenti: A, B, C, D, E, H, L

---

**Ld r, r'**

Carica il registro ad otto bit r con il contenuto del registro ad otto bit r'.

*L'istruzione non ha alcun effetto sul registro dei flag.*

**Esempio****LD H, A****Prima****Dopo**

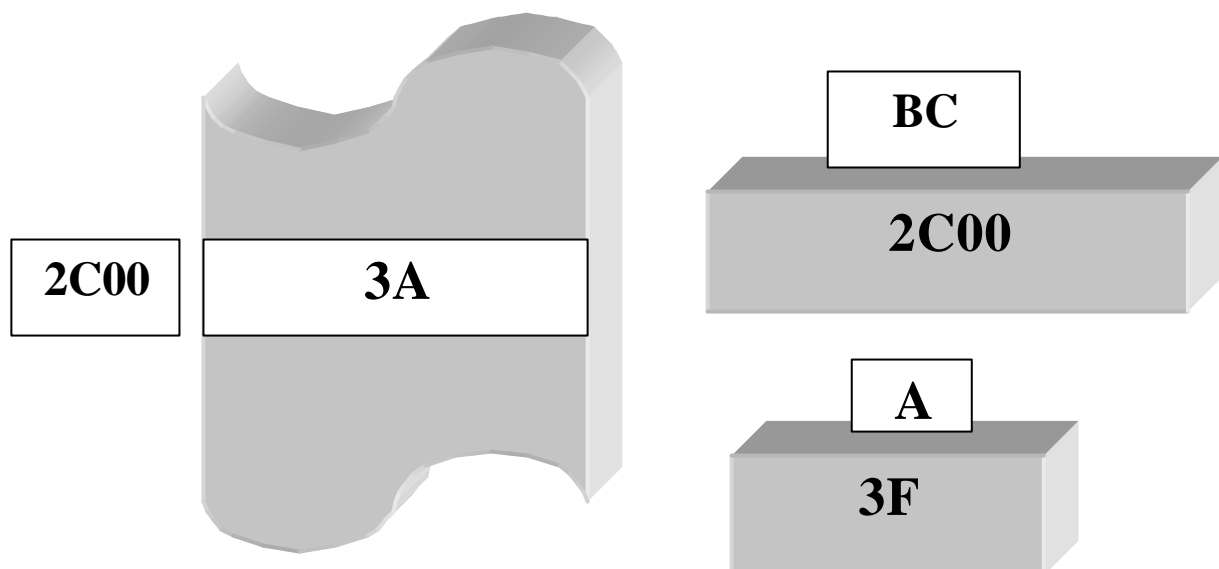
IL registro r ed il registro r' possono essere uno dei seguenti: A, B, C, D, E, H,

L

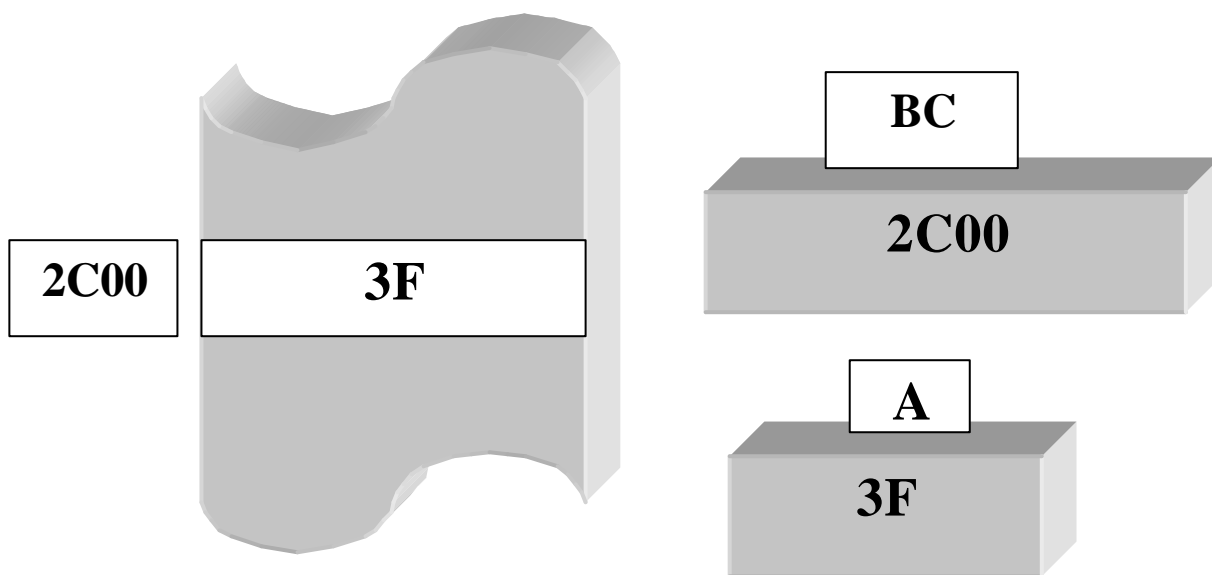
**LD (BC), A**

IL contenuto dell'accumulatore viene caricato in una locazione di memoria indirizzata in modo indiretto poiché il suo indirizzo è contenuto nel registro BC. Da notare che il registro BC si può usare per realizzare l'indirizzamento indiretto soltanto nei LOAD.

*L'istruzione non ha alcun effetto sul registro dei flag.*

**Esempio:****LD (BC), A****Prima**

**Dopo**



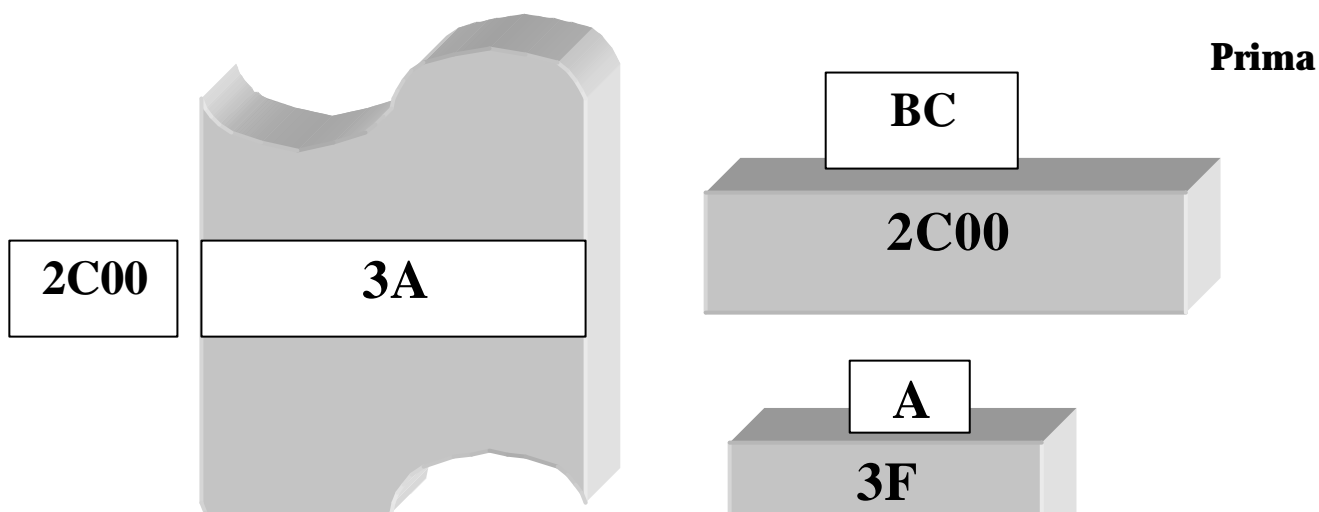
**LD A, (BC)**

L'accumulatore viene caricato con il contenuto di una locazione di memoria indirizzata in modo indiretto poiché il suo indirizzo è contenuto nel registro BC. Da notare che il registro BC si può usare per realizzare l'indirizzamento indiretto soltanto nei LOAD.

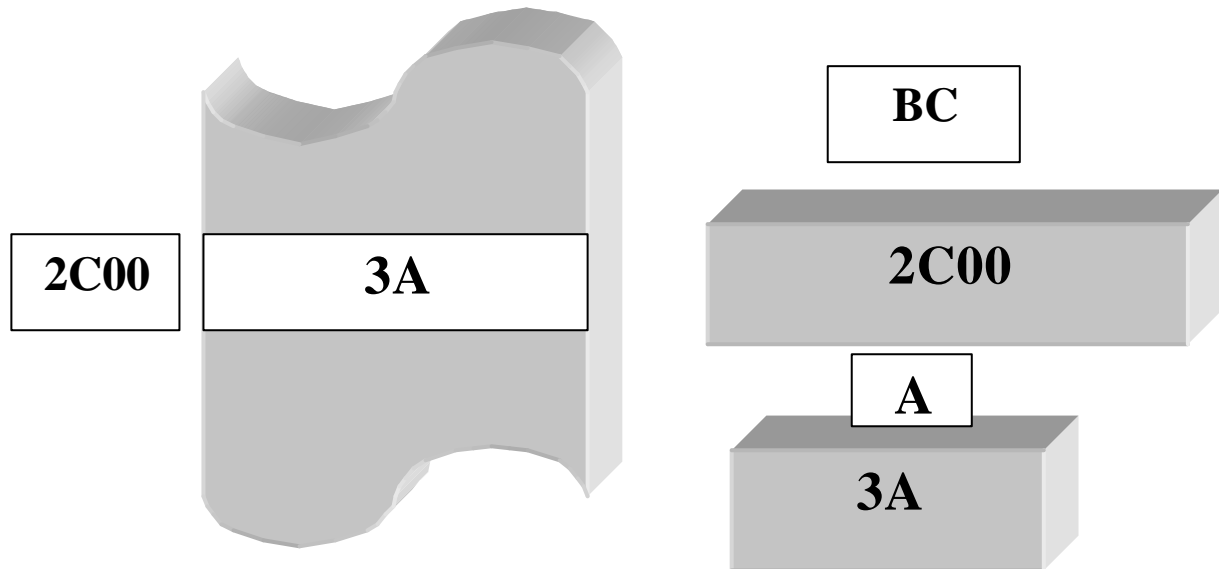
*L'istruzione non ha alcun effetto sul registro dei flag.*

**Esempio:**

**LD A, (BC)**



**Dopo**



**LD (DE), A**

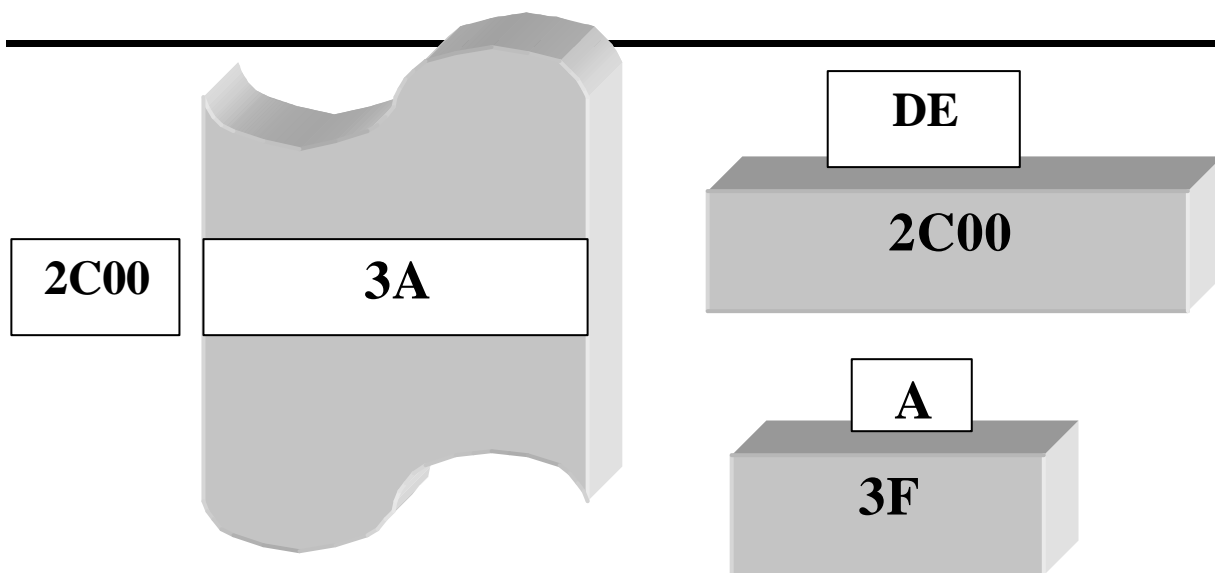
IL contenuto dell'accumulatore viene caricato in una locazione di memoria indirizzata in modo indiretto poiché il suo indirizzo è contenuto nel registro DE. Da notare che il registro DE si può usare per realizzare l'indirizzamento indiretto soltanto nei LOAD.

*L'istruzione non ha alcun effetto sul registro dei flag.*

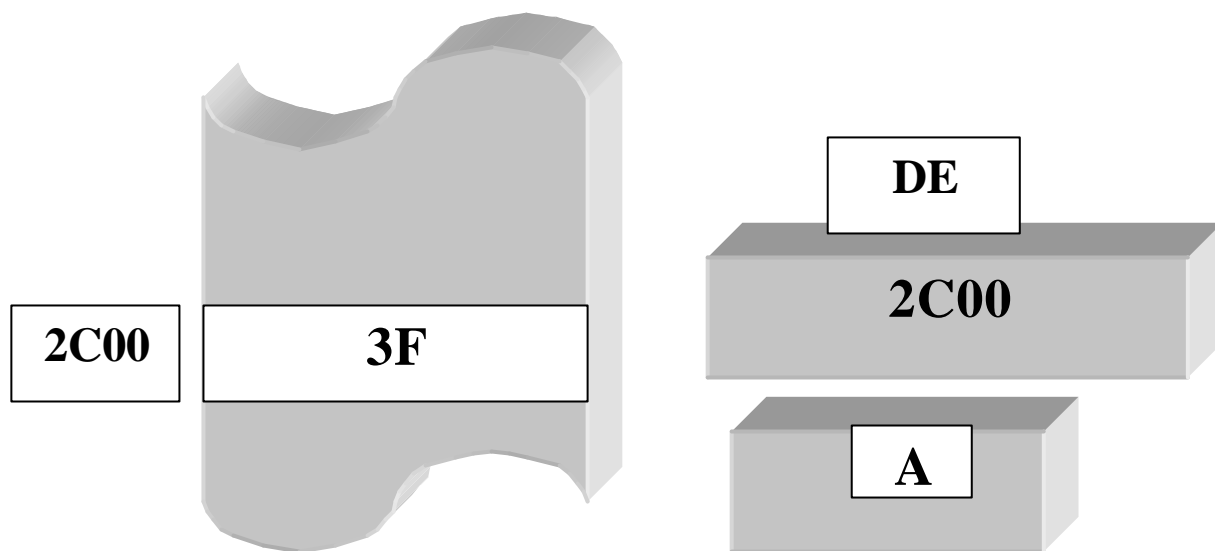
**Esempio:**

**LD (DE), A**

**Prima**



**Do-  
po**



**LD A, (DE)**

L'accumulatore viene caricato con il contenuto di una locazione di memoria indirizzata in modo indiretto poiché il suo indirizzo è contenuto nel registro

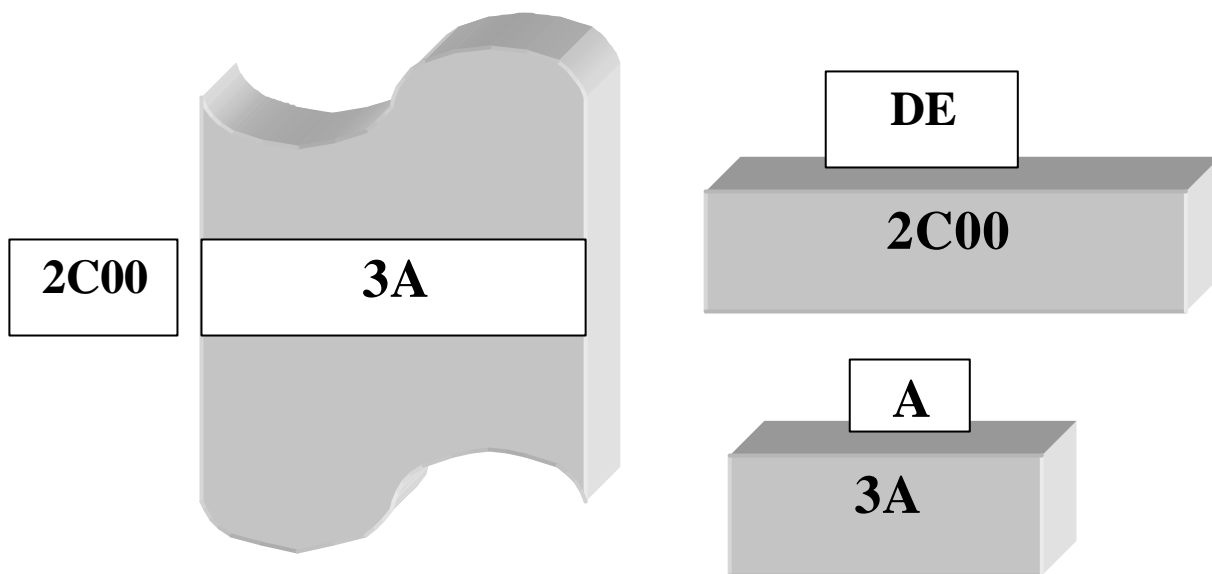
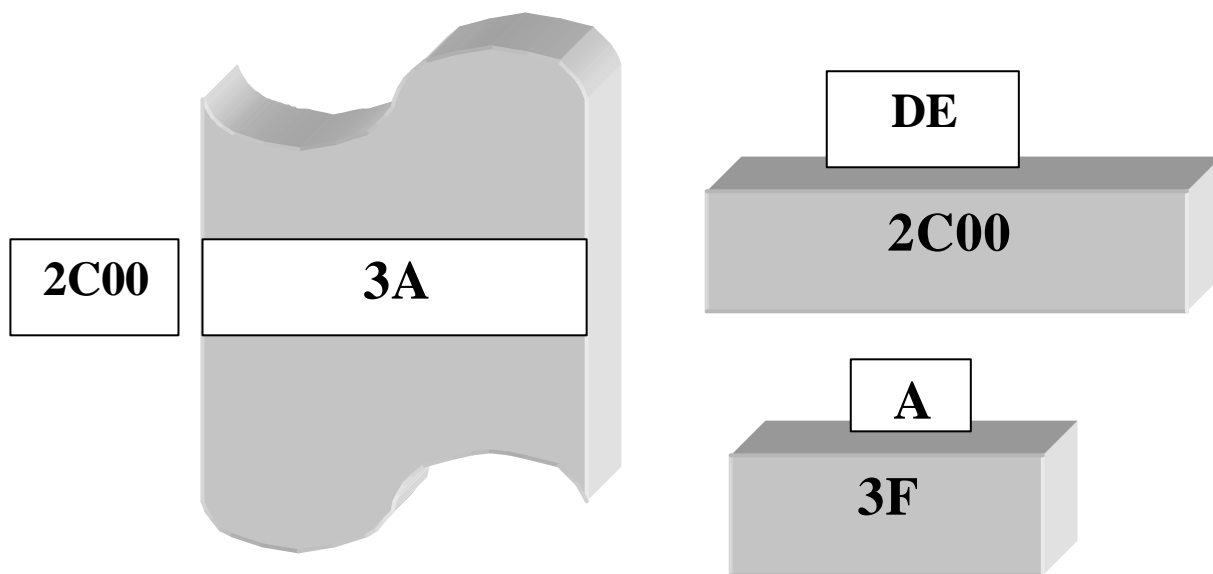
DE. Da notare che il registro DE si può usare per realizzare l'indirizzamento indiretto soltanto nei LOAD.

*L'istruzione non ha alcun effetto sul registro dei flag.*

**Esempio:**

**LD A, (DE)**

**Pri  
ma**



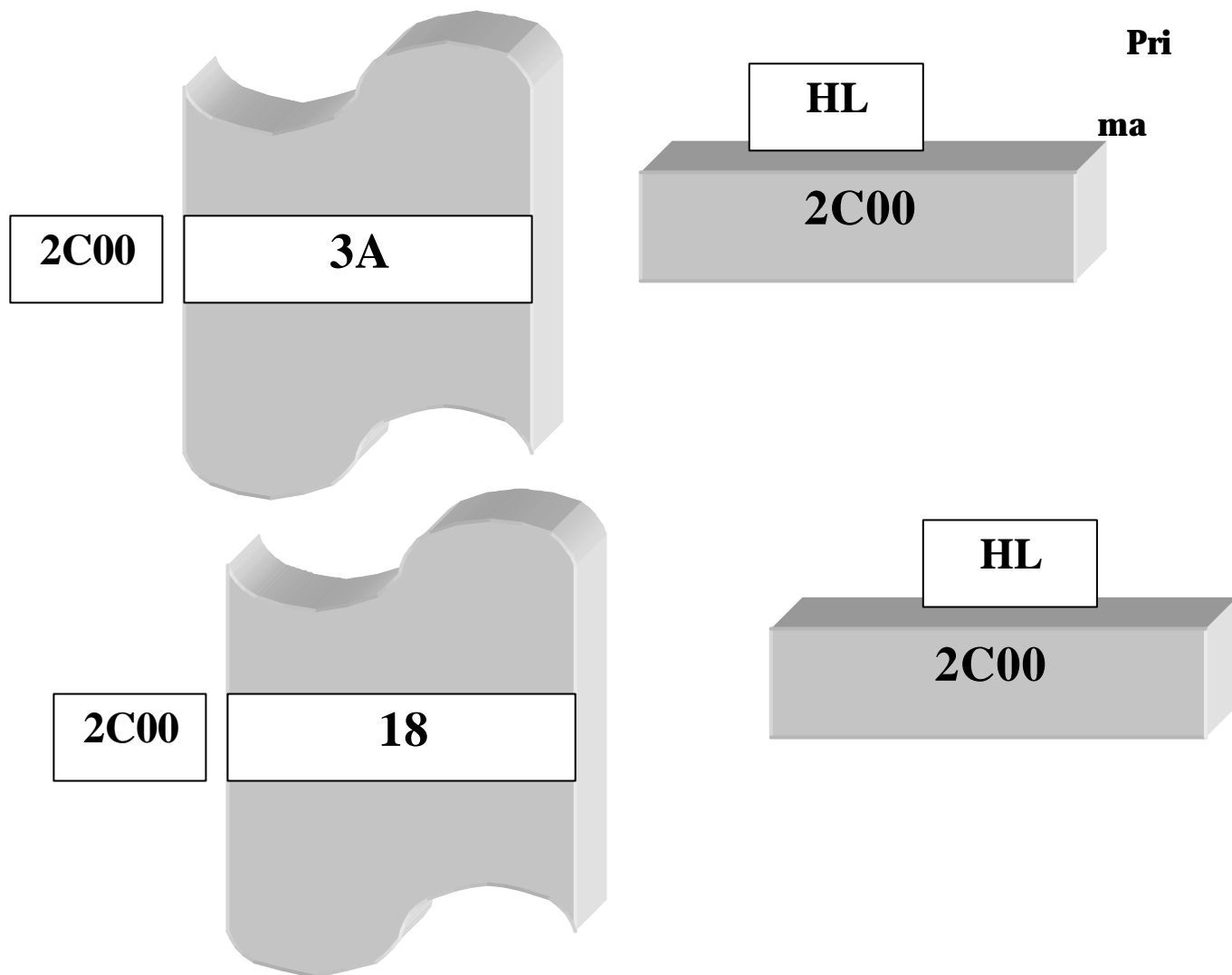
**Dopo**

**LD (HL), n**

La locazione di memoria il cui indirizzo è contenuto nel registro HL (indirizzamento indiretto) viene caricata con il dato ad otto bit n espresso direttamente nell'istruzione.

**Esempio:**

**LD (HL), 18h**



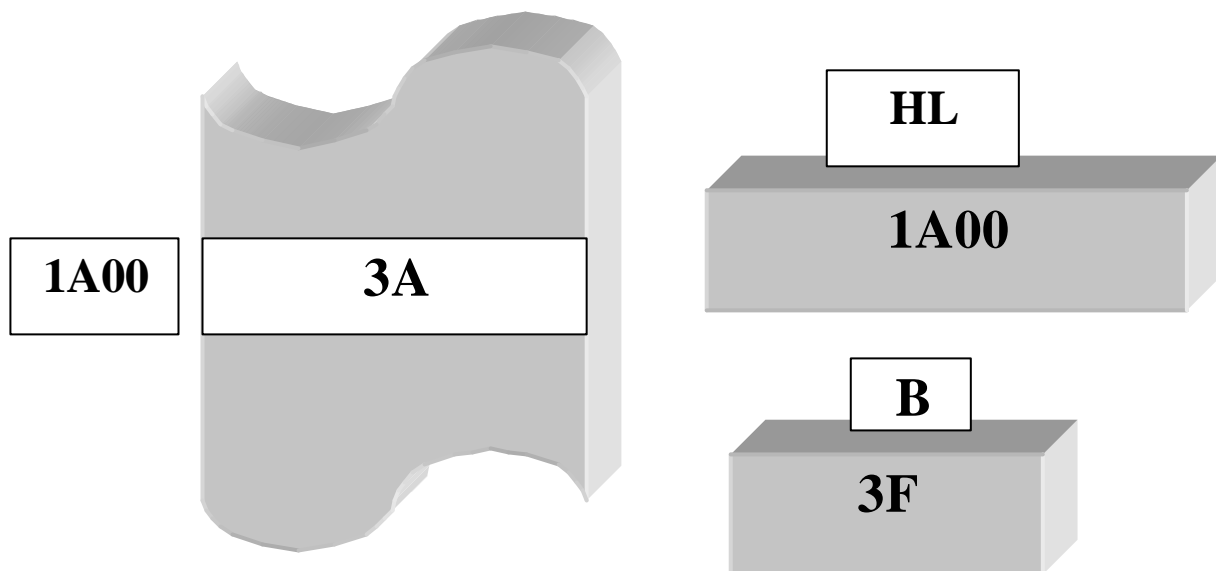
**Dopo**

**LD (HL), r**

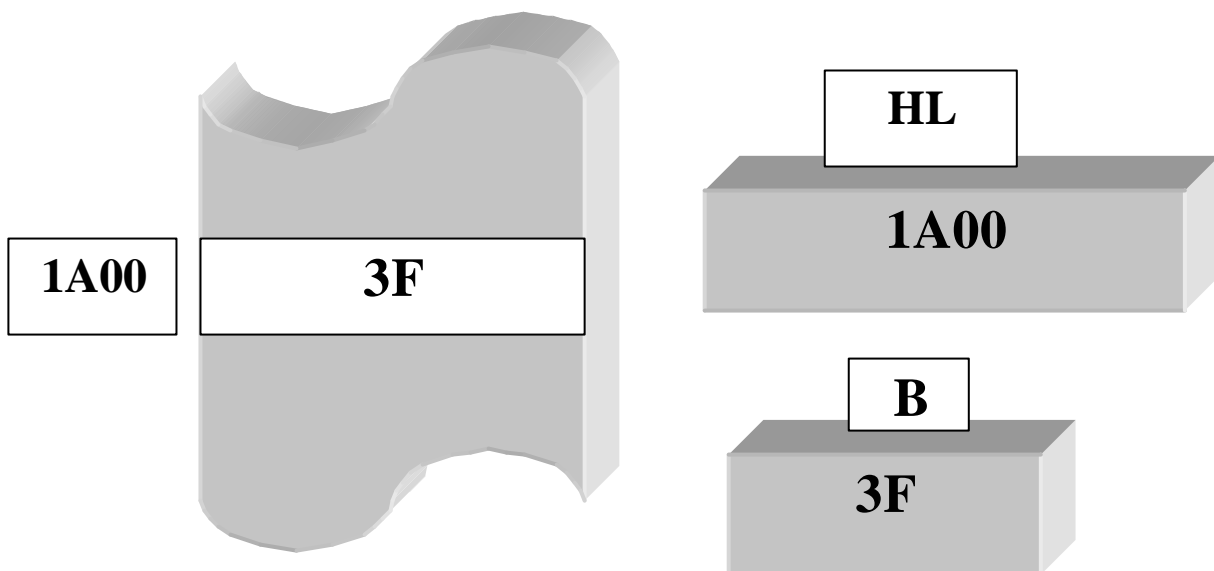
La locazione di memoria il cui indirizzo è contenuto nel registro HL (indirizzamento indiretto) viene caricata con il dato ad otto bit contenuto nel registro r.

**Esempio: LD (HL), B**

**Pri-  
ma**

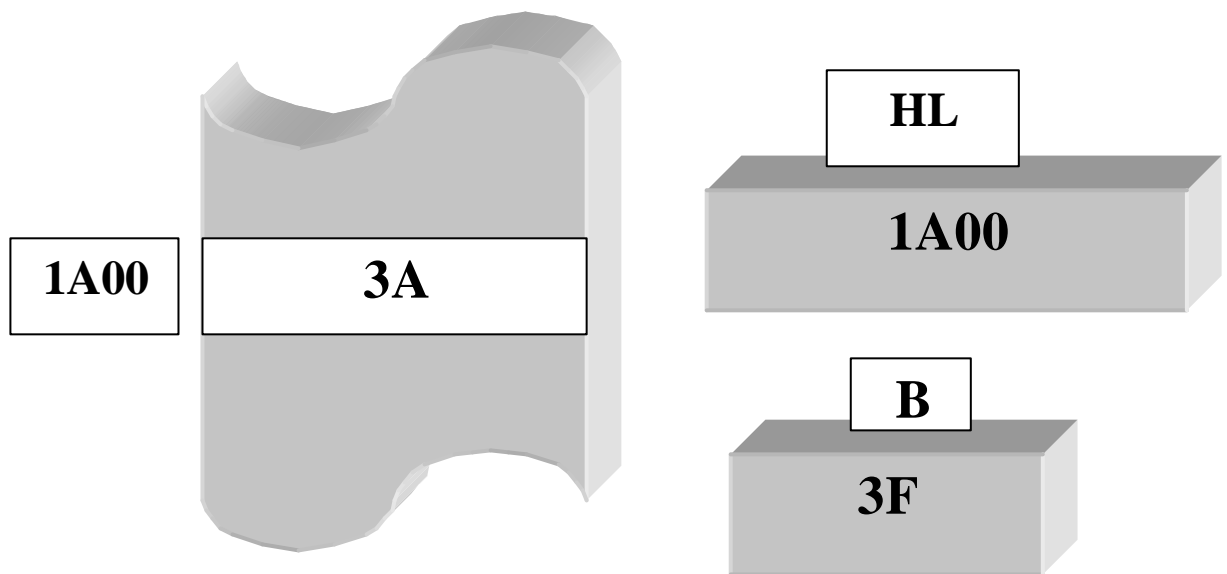
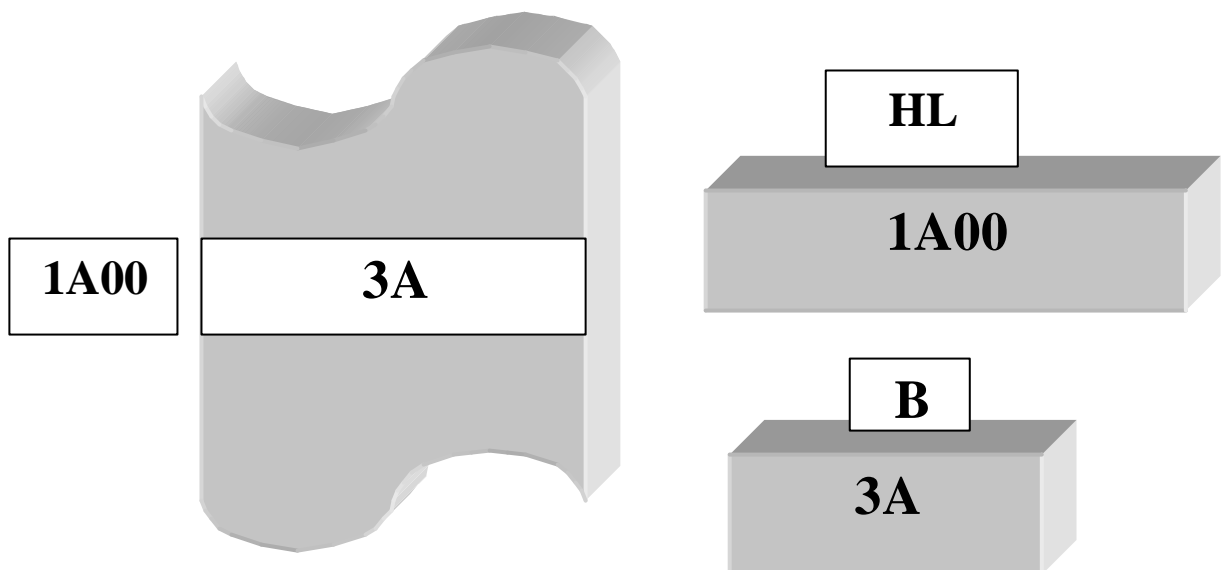


**D  
op  
o**



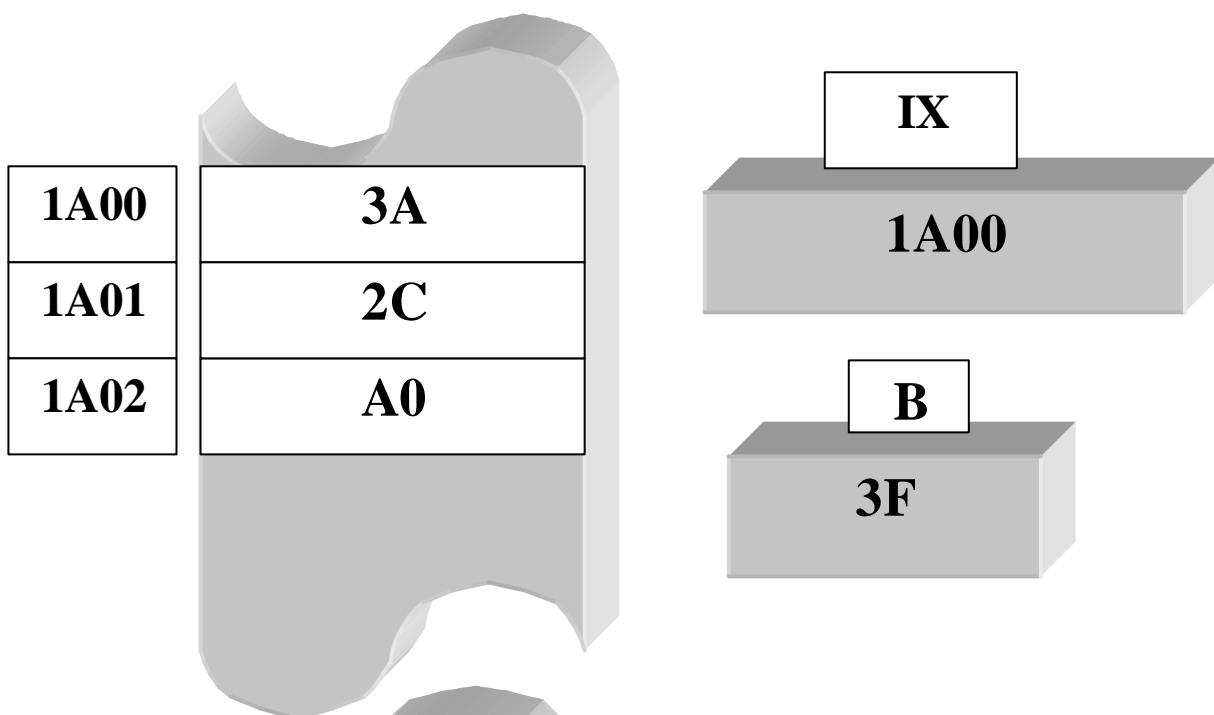
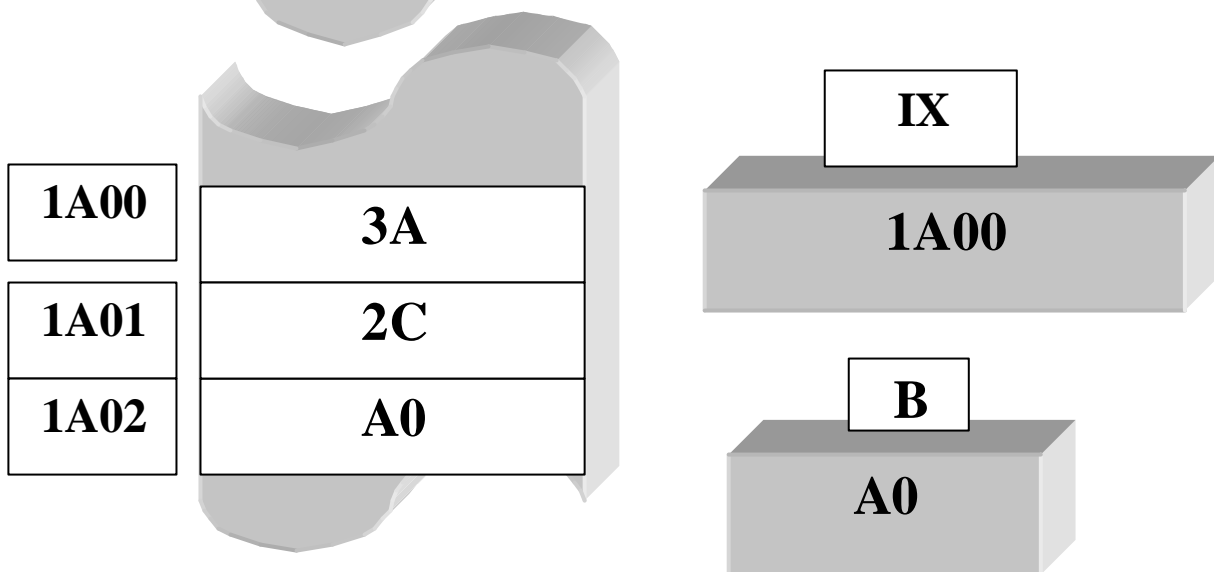
**LD r, (HL)**

Il registro ad otto bit r viene caricato con la locazione di memoria il cui indirizzo è contenuto nel registro HL (indirizzamento indiretto).

**Esempio: LD B, (HL)****Prima****Dopo**

**LD r, (IX+d)**

Il registro ad otto bit r viene caricato con la locazione di memoria il cui indirizzo si ottiene sommando il contenuto nel registro IX con lo piazzamento d (indirizzamento indicizzato).

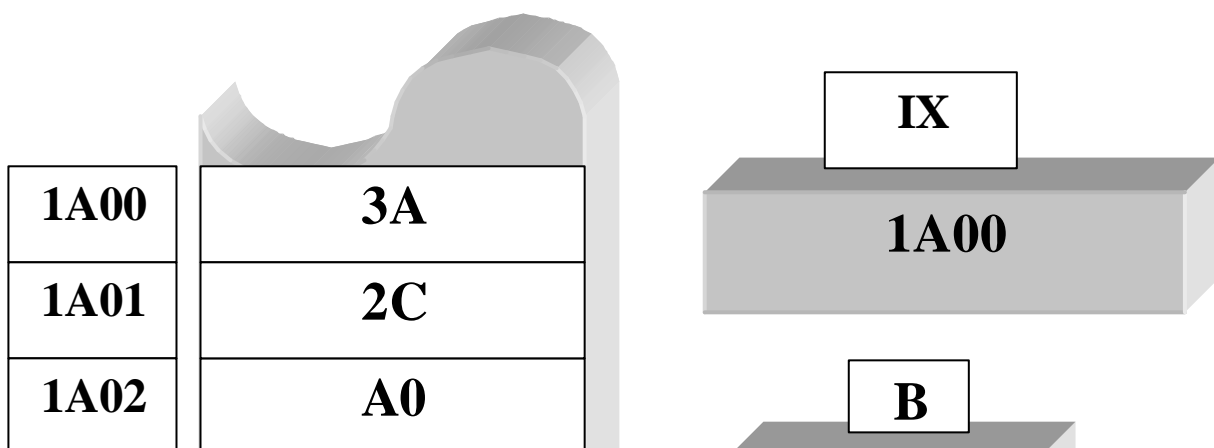
**Esempio:****LD B, (IX+2)****Prima****Dopo**

**LD (IX+d), r**

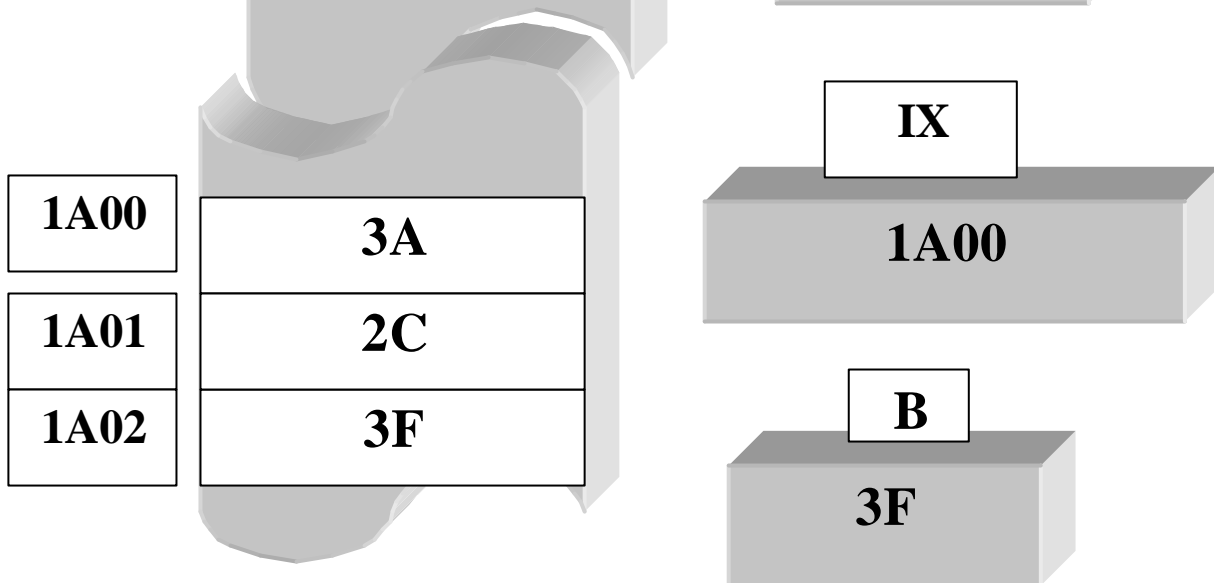
La locazione di memoria il cui indirizzo si ottiene sommando il contenuto del registro IX e lo spiazzamento d (indirizzamento indicizzato) viene caricata con il contenuto del registro ad otto bit r

**Esempio: LD (IX+2), B**

**Prima**



**Dopo**

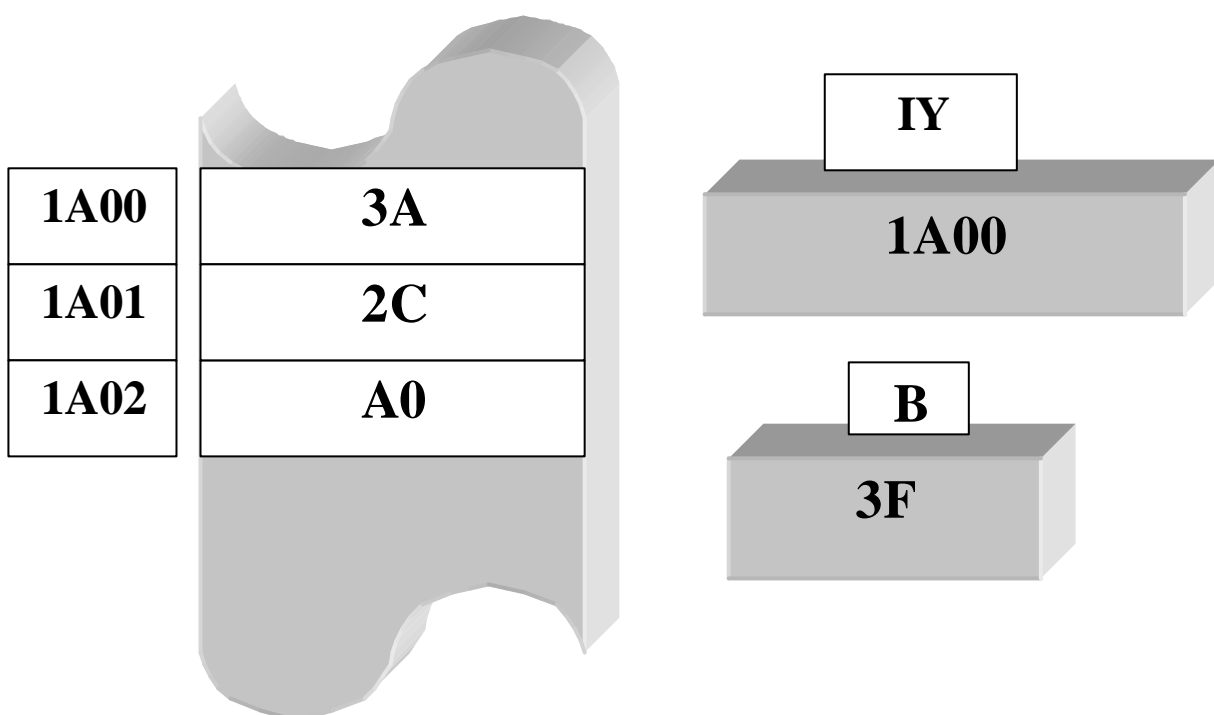


**LD r, (IY+d)**

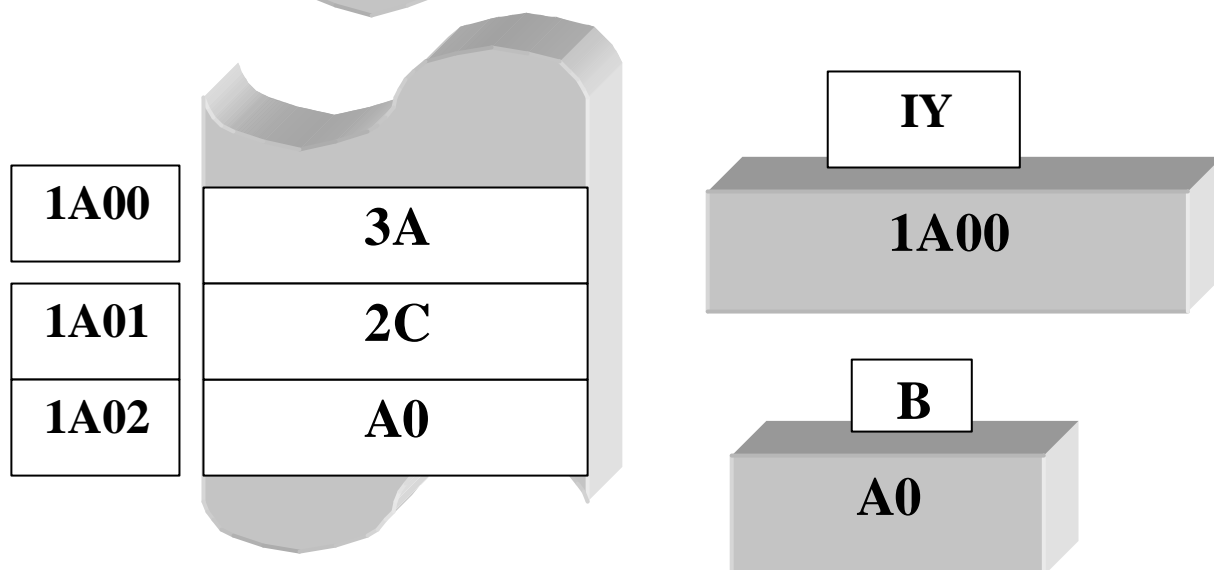
Il registro ad otto bit r viene caricato con la locazione di memoria il cui indirizzo si ottiene sommando il contenuto nel registro IY con lo piazzamento d (indirizzamento indicizzato).

**Esempio: LD B, (IY+2)**

**Prima**



**Dopo**



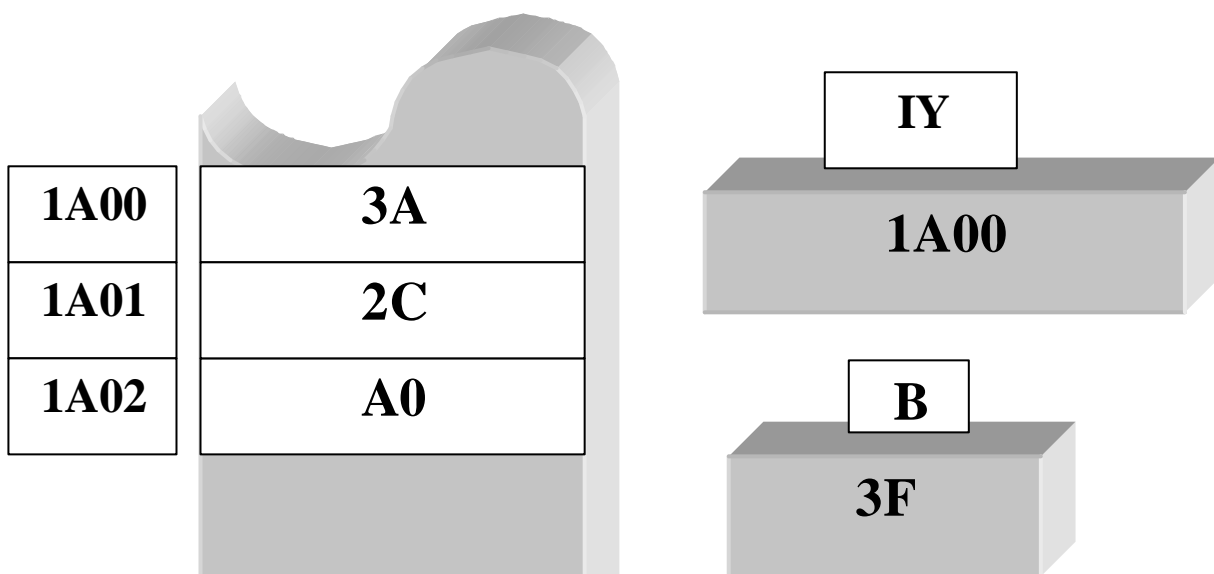
---

**LD (IY+d), r**

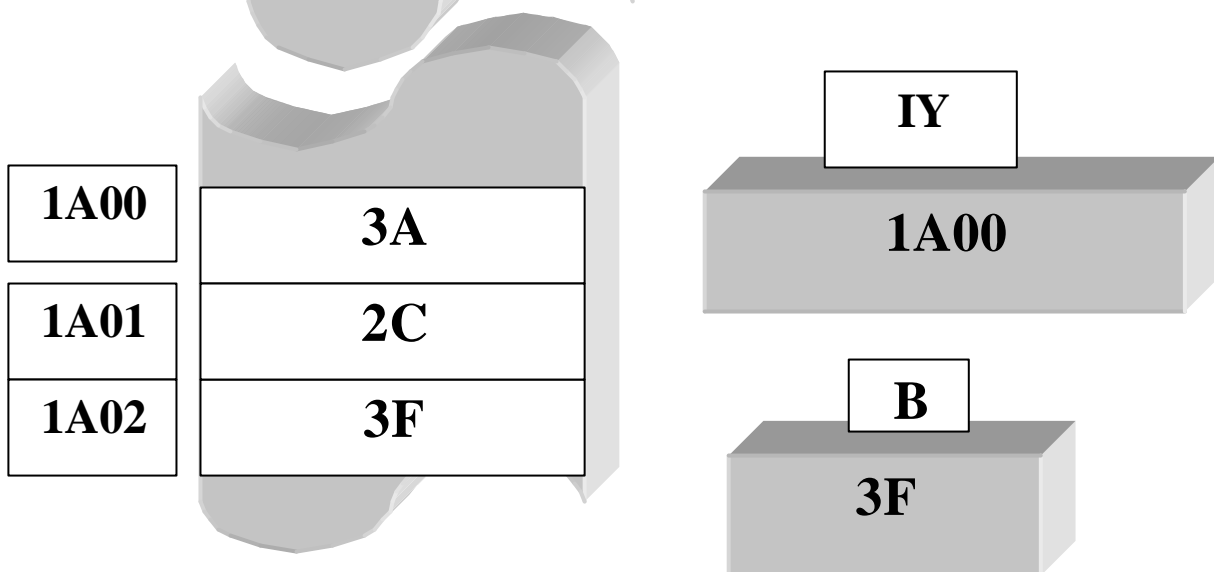
La locazione di memoria il cui indirizzo si ottiene sommando il contenuto del registro IY e lo spiazzamento d (indirizzamento indicizzato) viene caricata con il contenuto del registro ad otto bit r

**Esempio: LD (IY+2), B**

**Prima**



**Dopo**



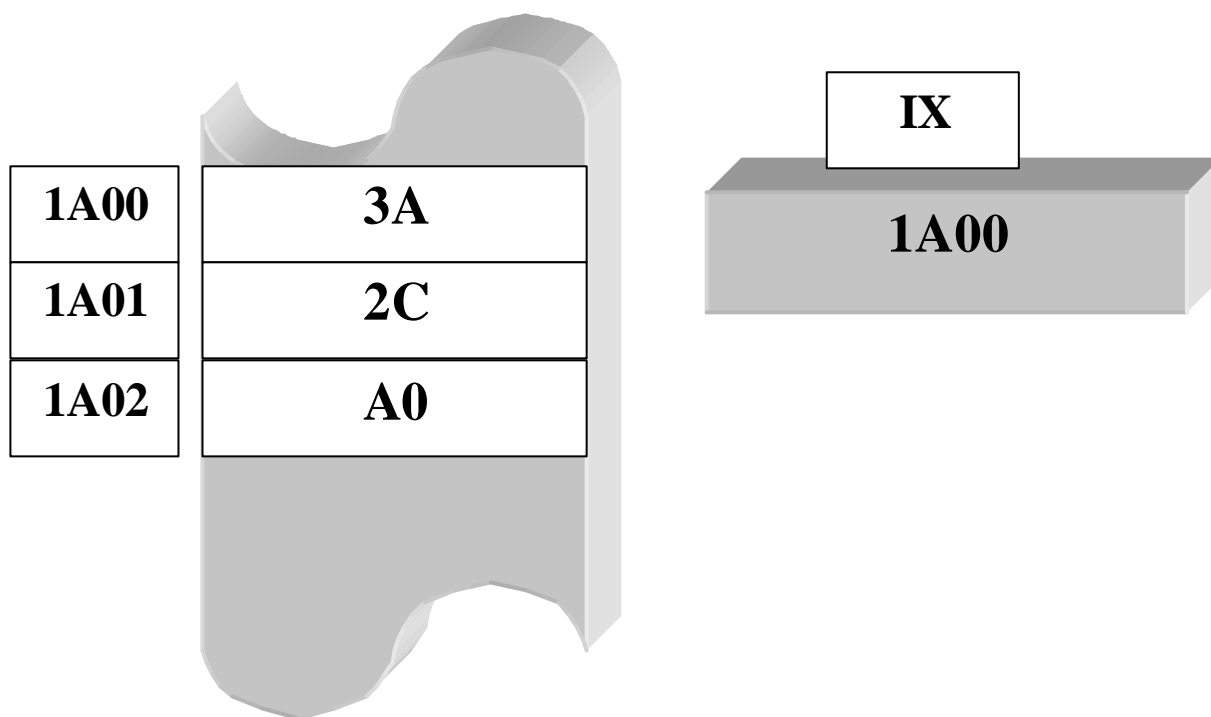
---

**LD (IX+d), n**

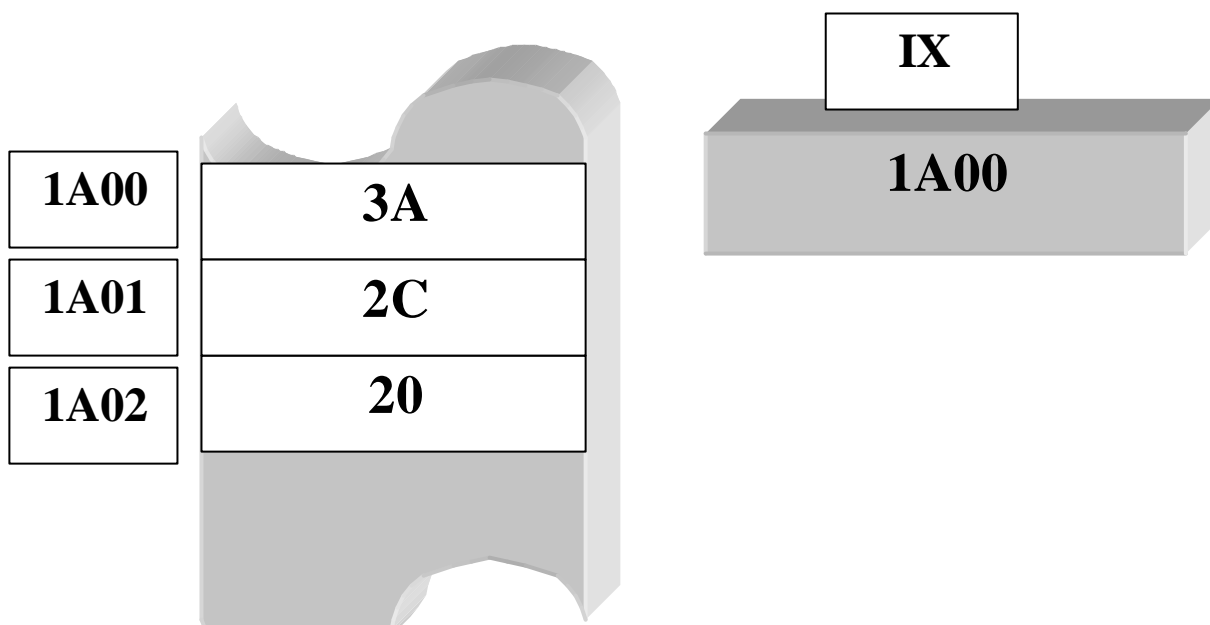
La locazione di memoria il cui indirizzo si ottiene sommando il contenuto nel registro IX con lo piazzamento d (indirizzamento indicizzato) viene caricata con il dato ad otto bit espresso immediatamente nell'istruzione.

**Esempio: LD (IX+2), 20h**

**Prima**

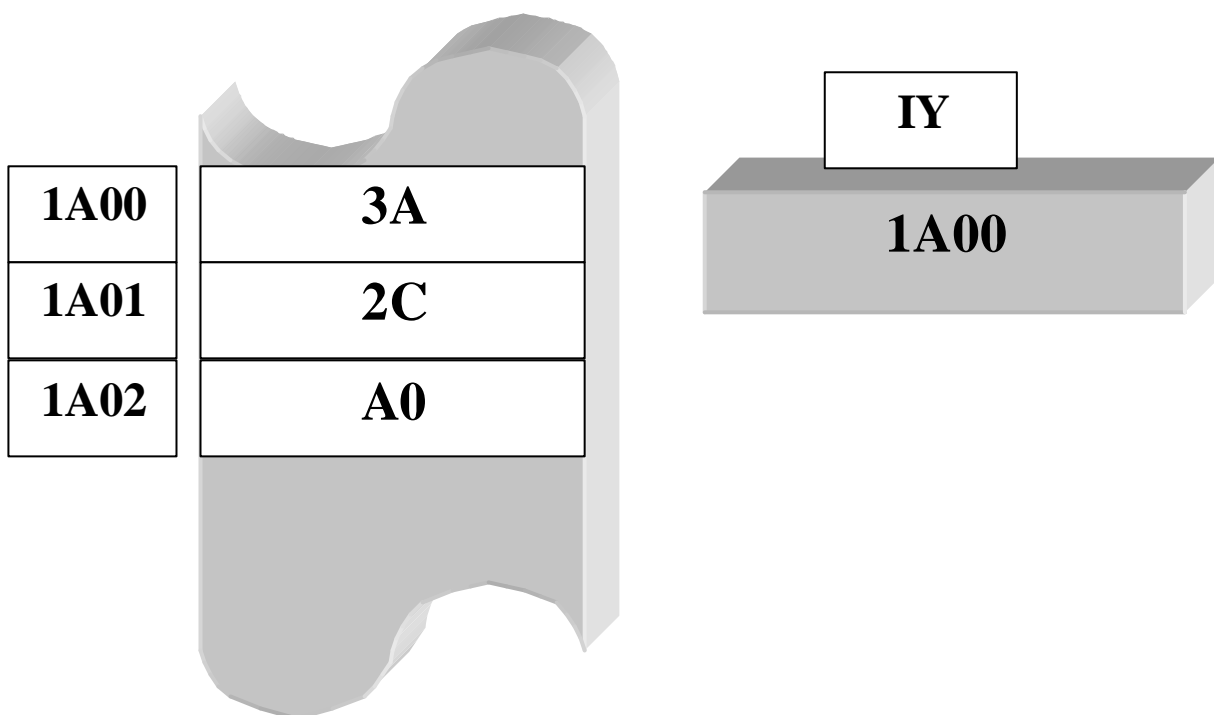
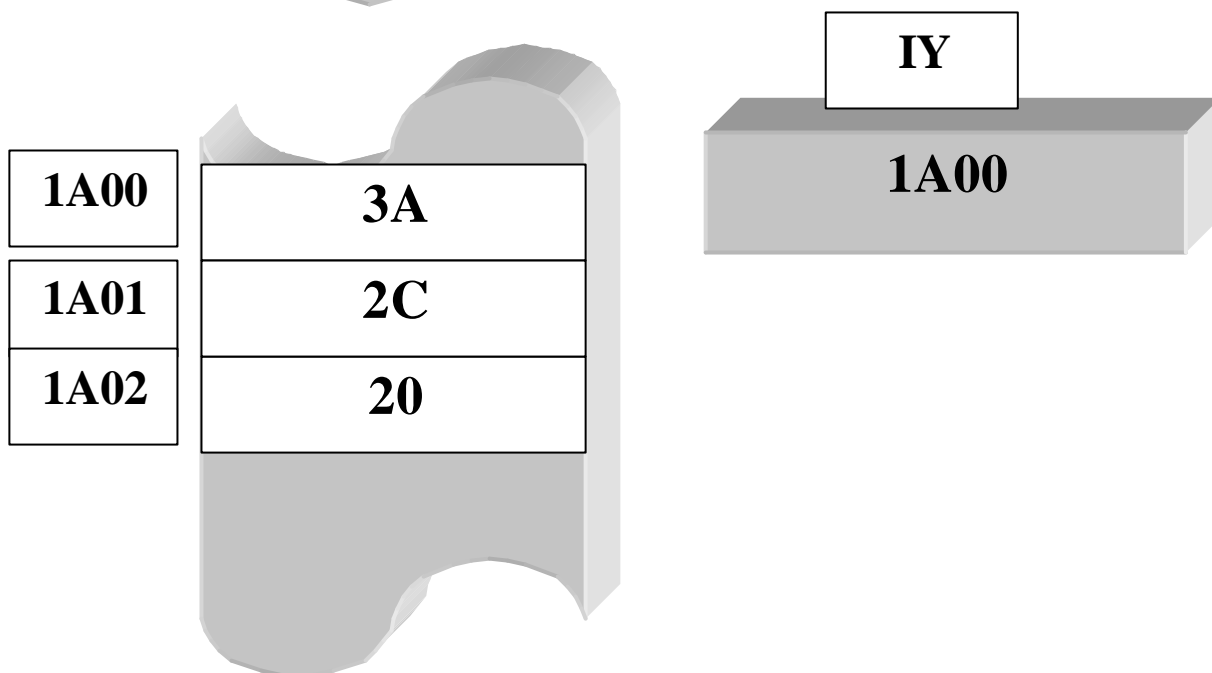


**Dopo**



**LD (IY+d), n**

La locazione di memoria il cui indirizzo si ottiene sommando il contenuto nel registro IY con lo piazzamento d (indirizzamento indicizzato) viene caricata con il dato ad otto bit espresso immediatamente nell'istruzione.

**Esempio:****LD (IY+2), 20h****Prima****Dopo**

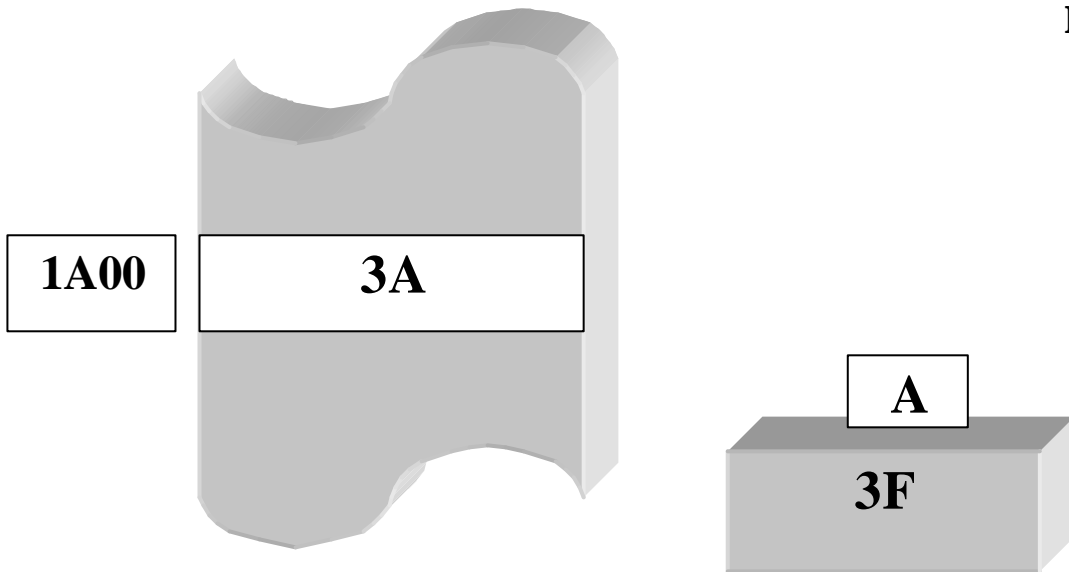
---

**LD A, (nn)**

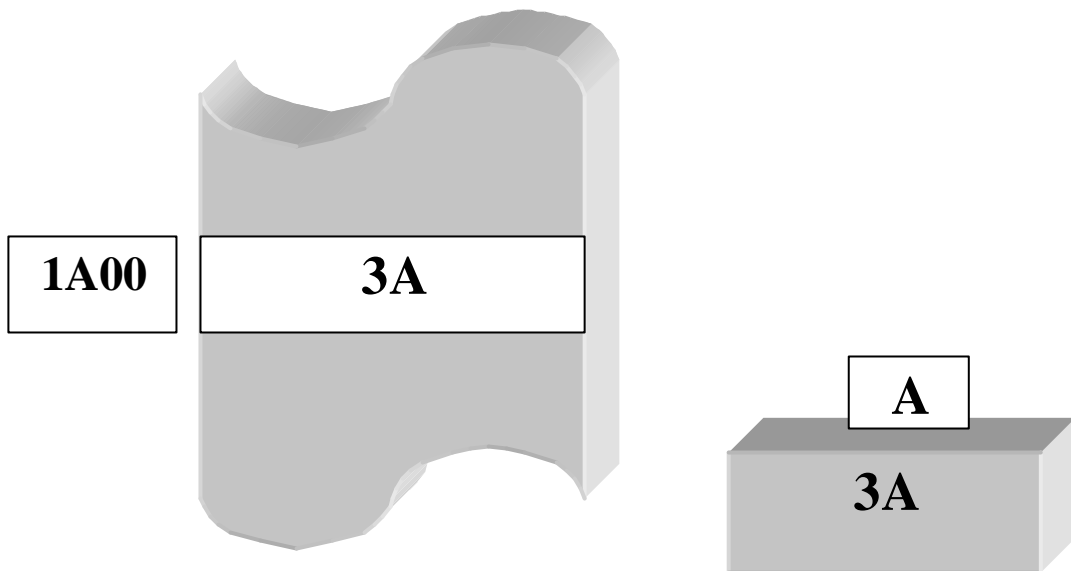
L'accumulatore viene caricato con il contenuto della locazione di memoria il cui indirizzo è espresso direttamente nell'istruzione (indirizzamento diretto).

**Esempio: LD A, (1A00)**

**Prima**



**Dopo**



---

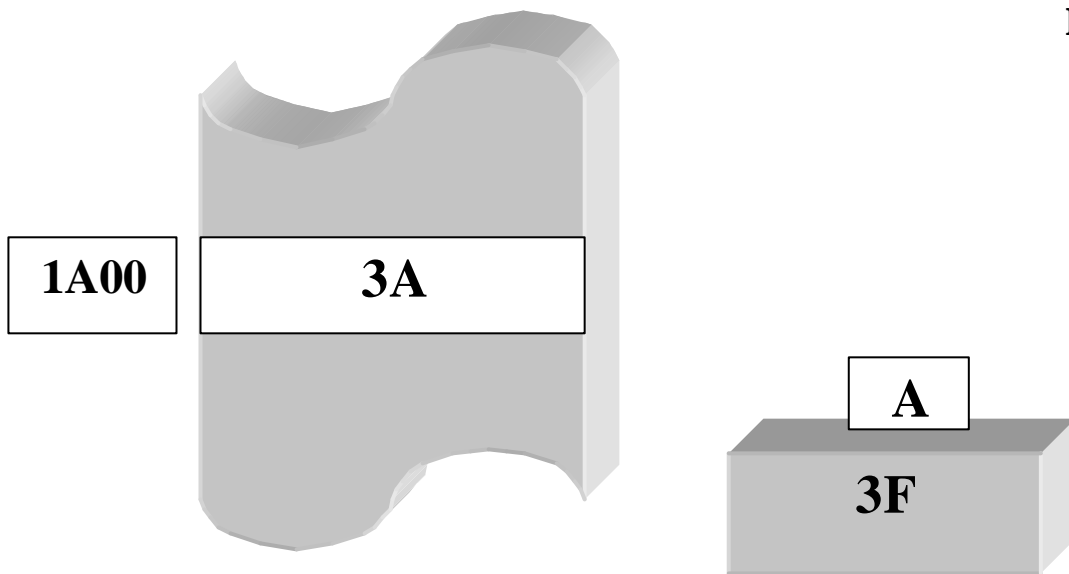
**LD (nn), A**

La locazione di memoria il cui indirizzo è espresso direttamente nell'istruzione (indirizzamento diretto), viene caricata con il contenuto dell'accumulatore.

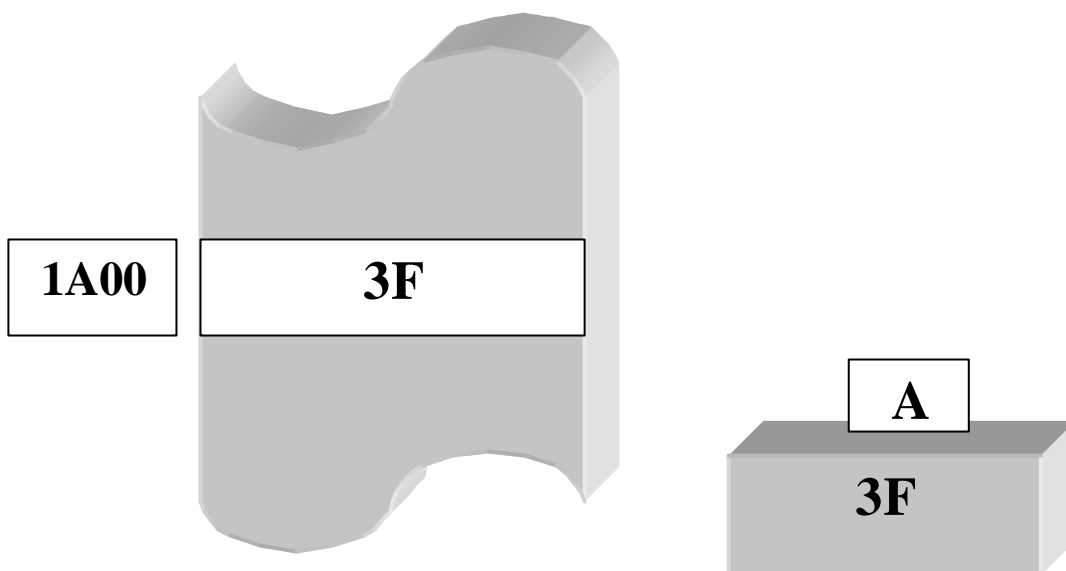
**Esempio:**

**LD (1A00), A**

**Prima**

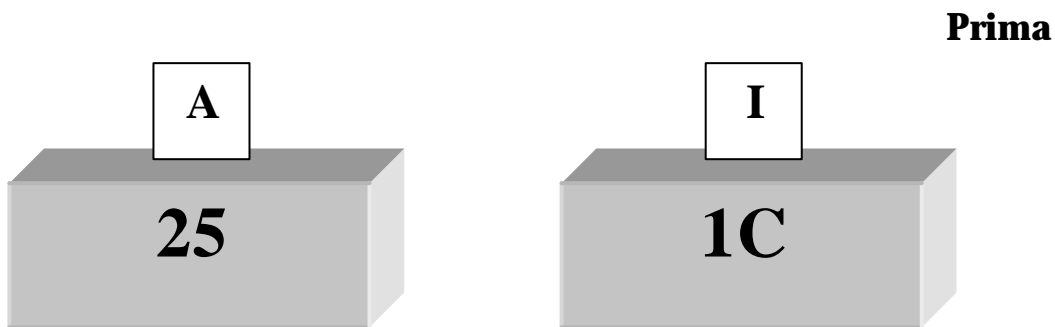


**Dopo**

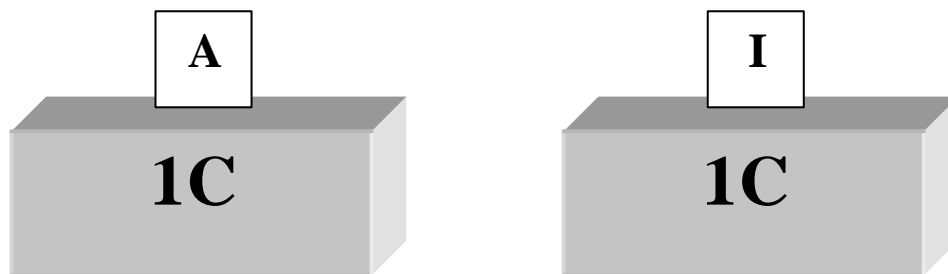


**LD A, I**

L'accumulatore viene caricato con il contenuto del registro I



**Dopo**

**Modifiche al registro dei flag**

**S**      **Z**                      **H**                      **P/V**      **N**                      **C**

●	●		<b>0</b>		<b>X</b>	<b>0</b>	
---	---	--	----------	--	----------	----------	--

Il flag S viene settato a seconda del segno del dato trasferito.

IL flag Z viene settato ad uno se il dato trasferito è nullo, altrimenti viene posto a zero.

I flag H ed N vengono posti sempre a zero

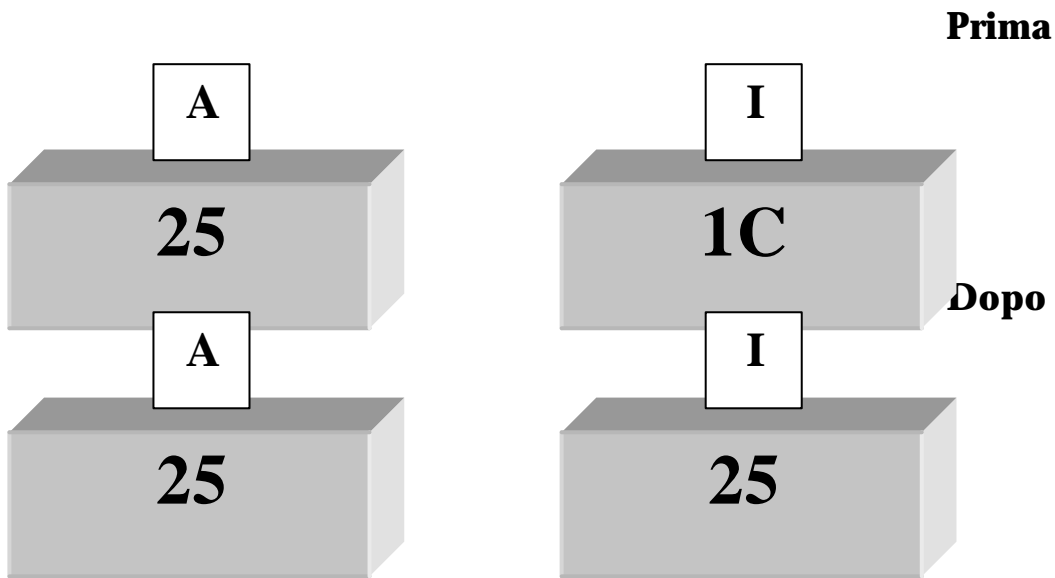
IL falg P/V assume il valore del registro IFF2

---

Gli altri flag non vengono modificati

**LD I, A**

IL registro I viene caricato con il contenuto dell'accumulatore

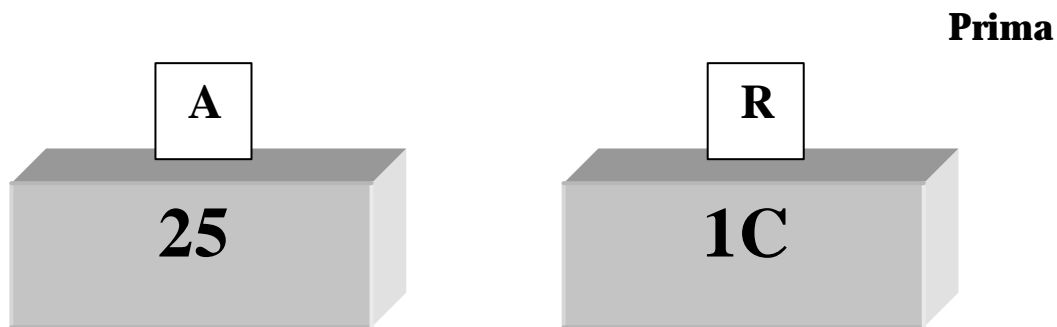


**Modifiche al registro dei flag**

Il registro dei flag non viene modificato

**LD A, R**

L'accumulatore viene caricato con il contenuto del registro R

**Modifiche al registro dei flag**

**S**      **Z**                      **H**                      **P/V**      **N**      **C**

●	●		<b>0</b>		<b>X</b>	<b>0</b>	
---	---	--	----------	--	----------	----------	--

Il flag S viene settato a seconda del segno del dato trasferito.

IL flag Z viene settato ad uno se il dato trasferito è nullo, altrimenti viene posto a zero.

I flag H ed N vengono posti sempre a zero

IL falg P/V assume il valore del registro IFF2

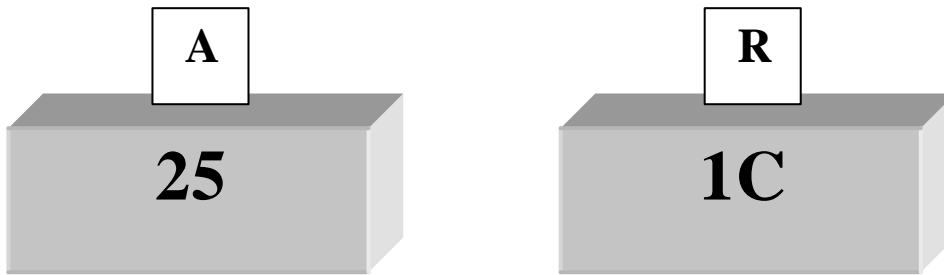
---

Gli altri flag non vengono modificati

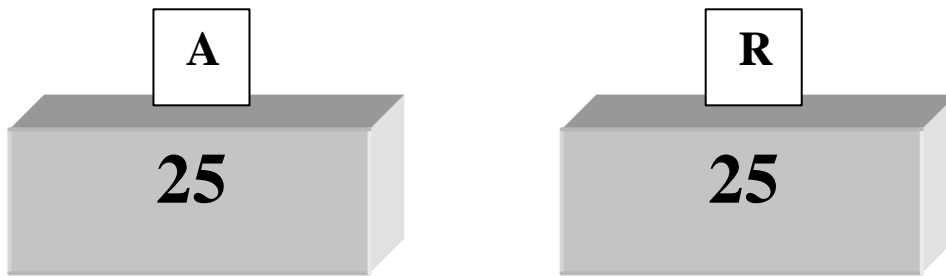
**LD R, A**

IL registro R viene caricato con il contenuto dell'accumulatore

**Prima**



**Dopo**



**Modifiche al registro dei flag**

Il registro dei flag non viene modificato

---

**LD SP, dd**

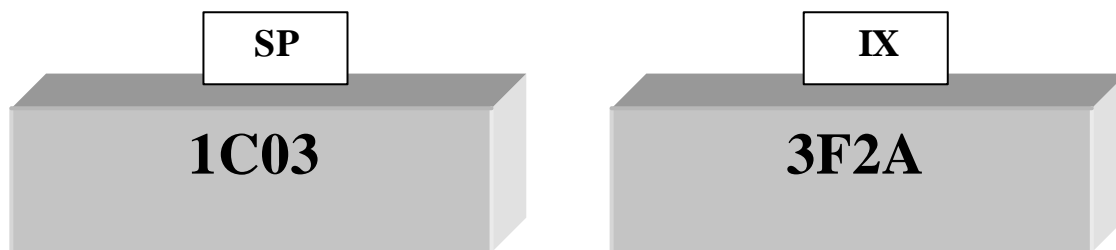
Lo Stack Pointer viene caricato con il contenuto del registro a sedici bit dd.

Il registro dd può essere uno dei seguenti: HL, IX, IY.

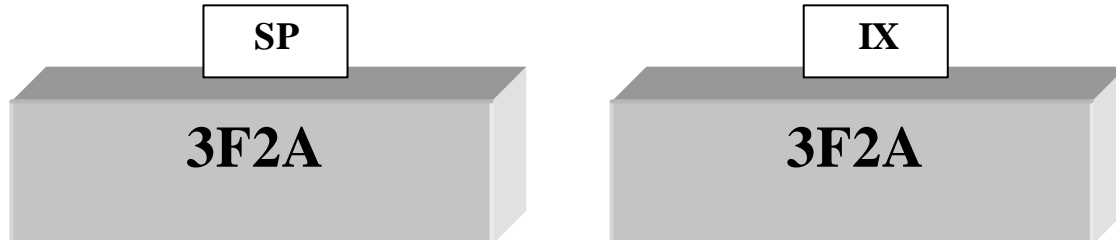
Non viene modificato il registro dei flag.

**Esempio: LD SP, IX**

**Prima**



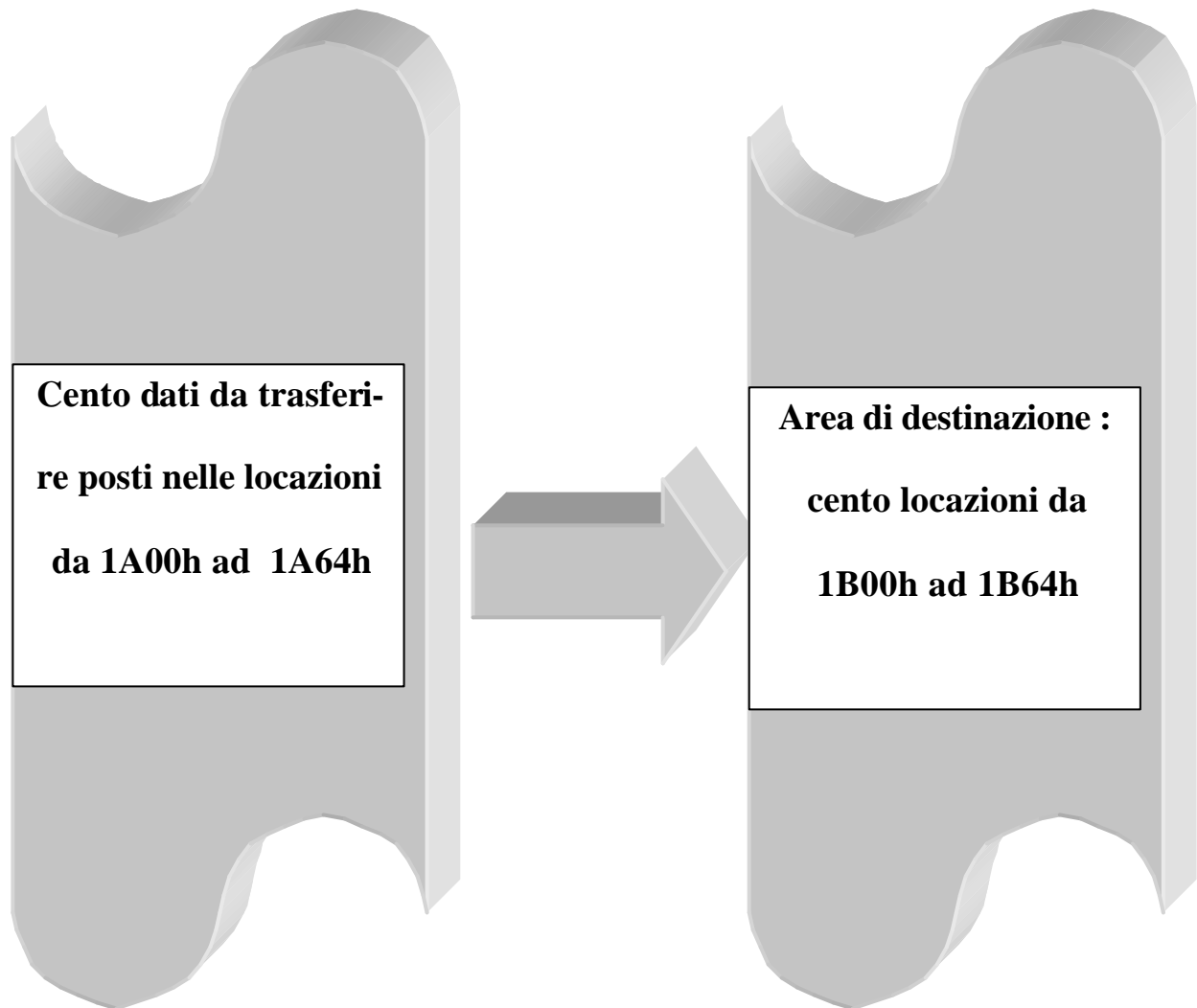
**Dopo**



---

**LDD**

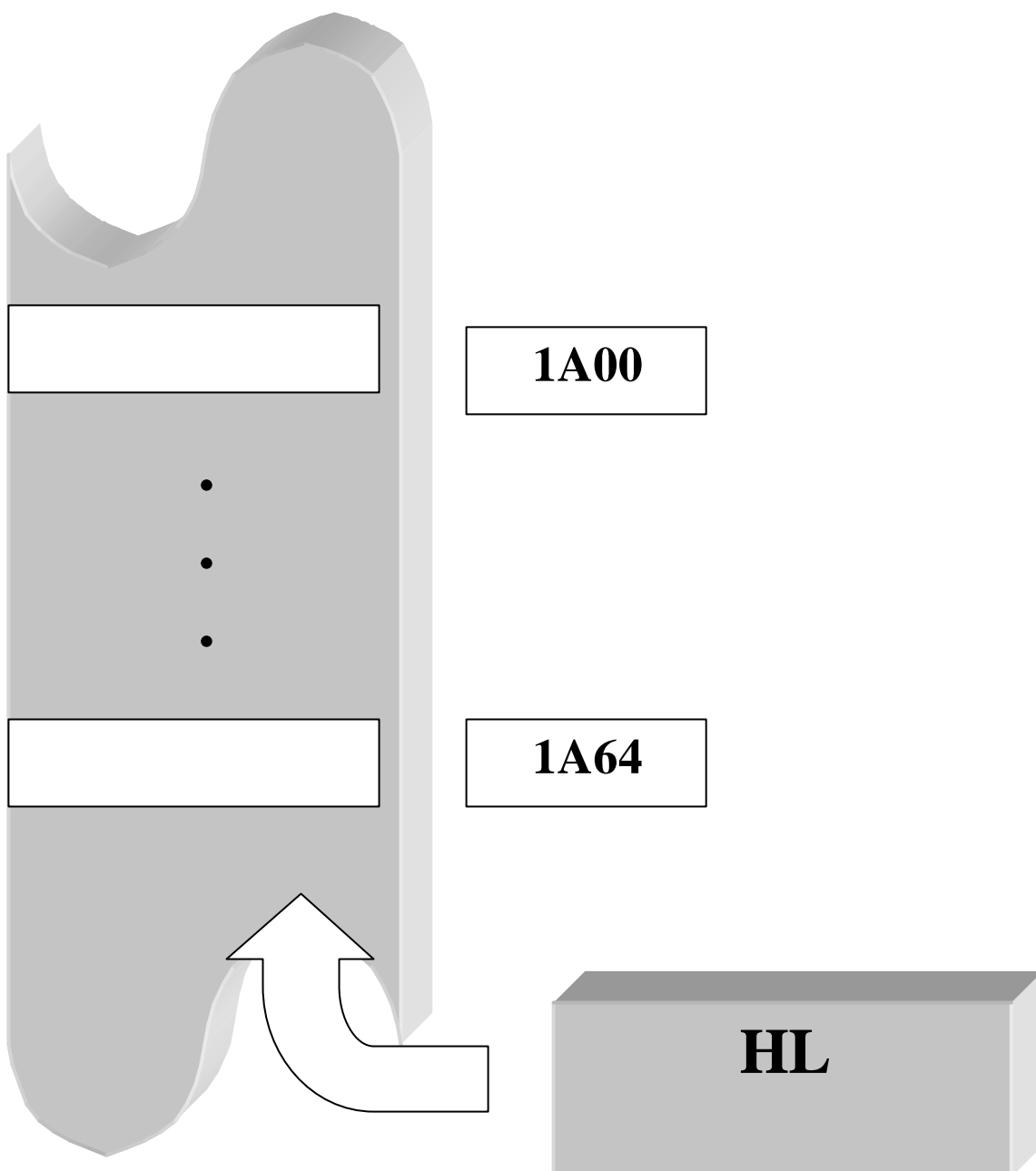
Questa istruzione viene utilizzata quando si vuole trasferire un blocco di dati da una zona all'altra della memoria

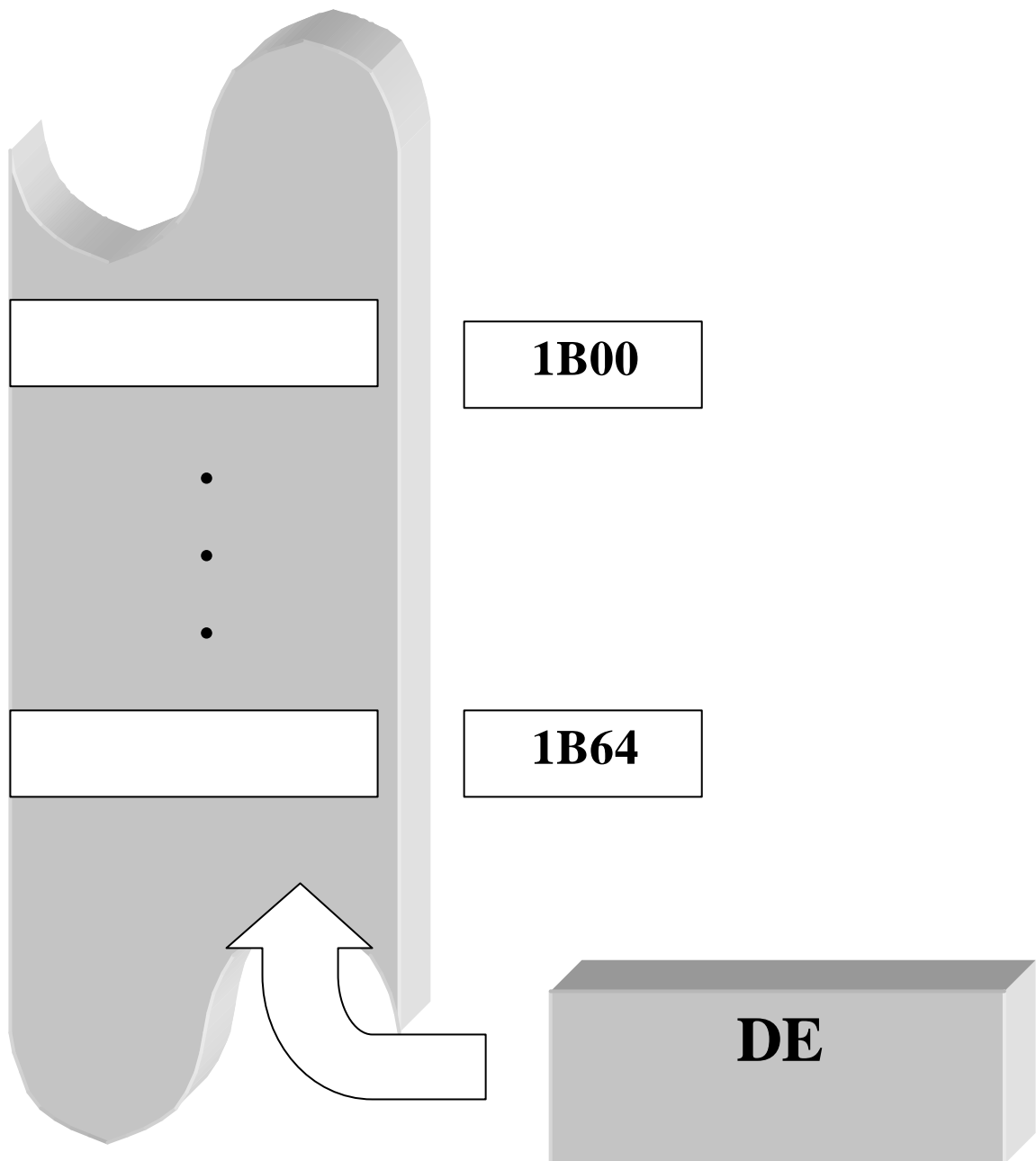


In questo caso occorrono

- un puntatore alla zona di memoria sorgente dei dati, cioè un registro a 16 bit contenente l'indirizzo della locazione da cui prelevare il dato che è il registro HL,

- un puntatore alla zona di memoria destinazione dei dati, cioè un registro a 16 bit contenente l'indirizzo della locazione in cui spostare il dato, che è il registro DE
- un registro che faccia da contatore per conoscere quando sarà stato terminato il processo di trasferimento dei dati, che è il registro BC





In questa istruzione si parte dalle locazioni di indirizzo più elevato e si prosegue verso le locazioni di indirizzo più basso.

Ogni volta che viene eseguita l'istruzione viene letto il dato contenuto nella locazione puntata da HL, viene trasferito nella locazione puntata da DE, vengono decrementati HL, DE ed il contatore BC.

**(DE) ← (HL)**

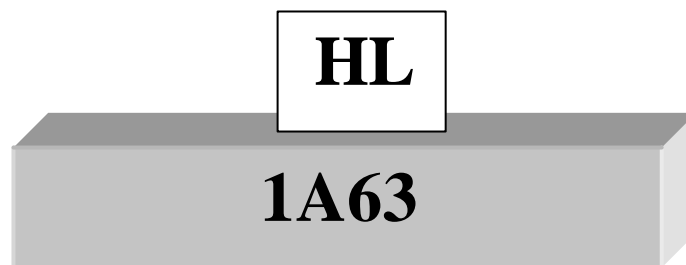
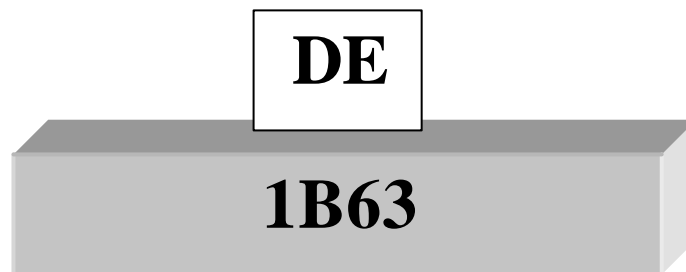
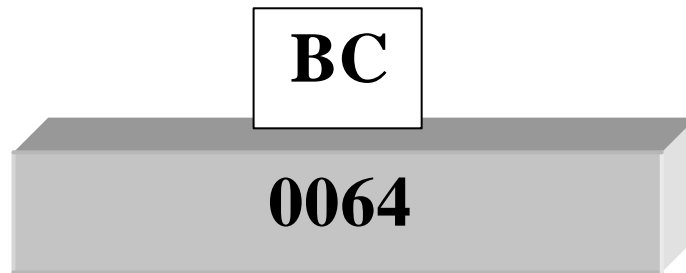
**DE=DE-1**

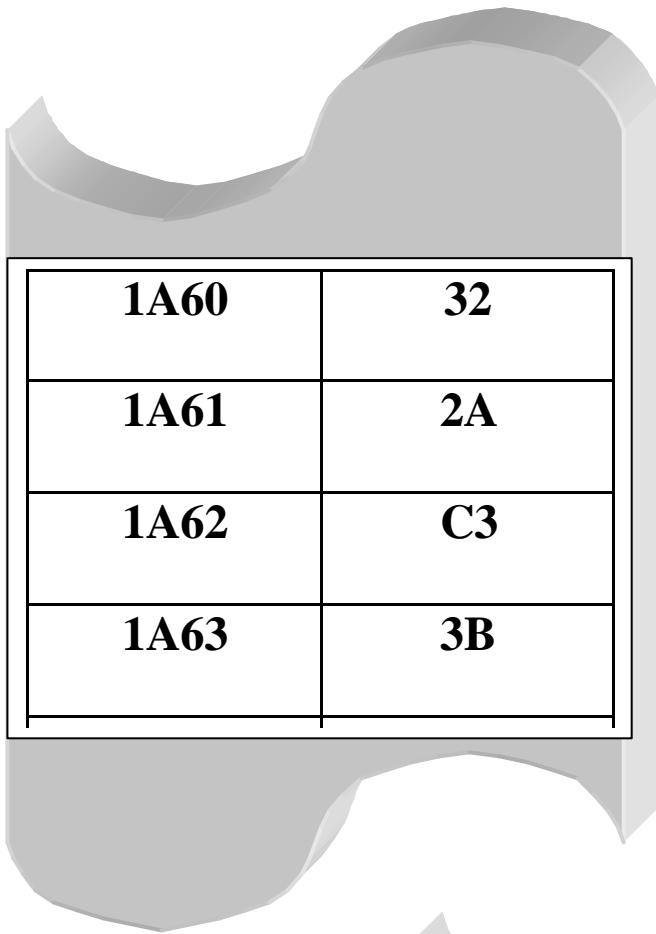
**HL=HL-1**

**BC=BC-1**

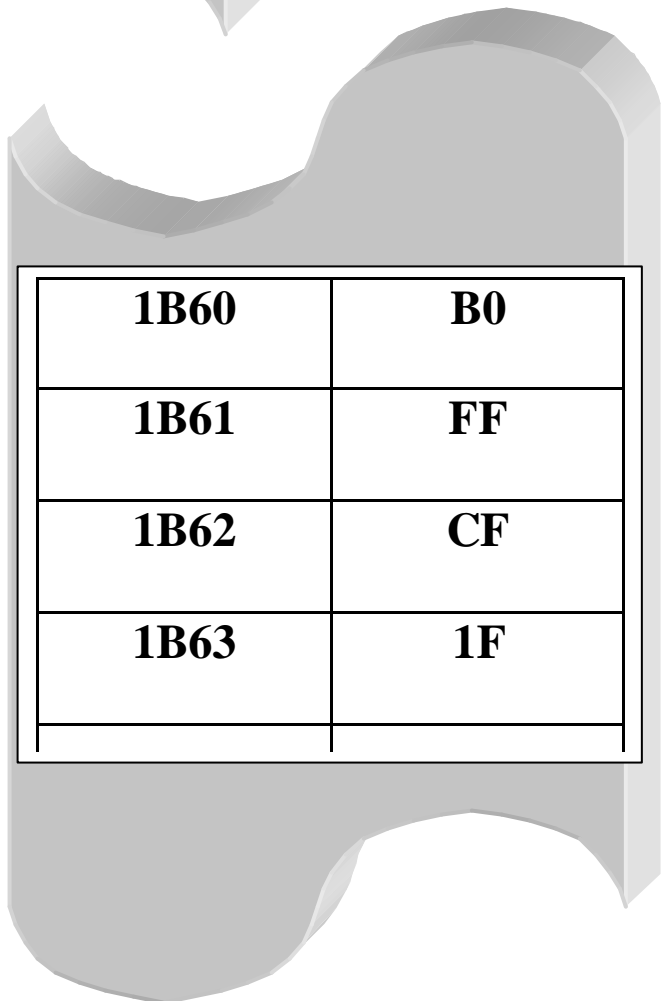
**Esempio**

**Prima**





<b>1A60</b>	<b>32</b>
<b>1A61</b>	<b>2A</b>
<b>1A62</b>	<b>C3</b>
<b>1A63</b>	<b>3B</b>



<b>1B60</b>	<b>B0</b>
<b>1B61</b>	<b>FF</b>
<b>1B62</b>	<b>CF</b>
<b>1B63</b>	<b>1F</b>

**Dopo**

**BC**

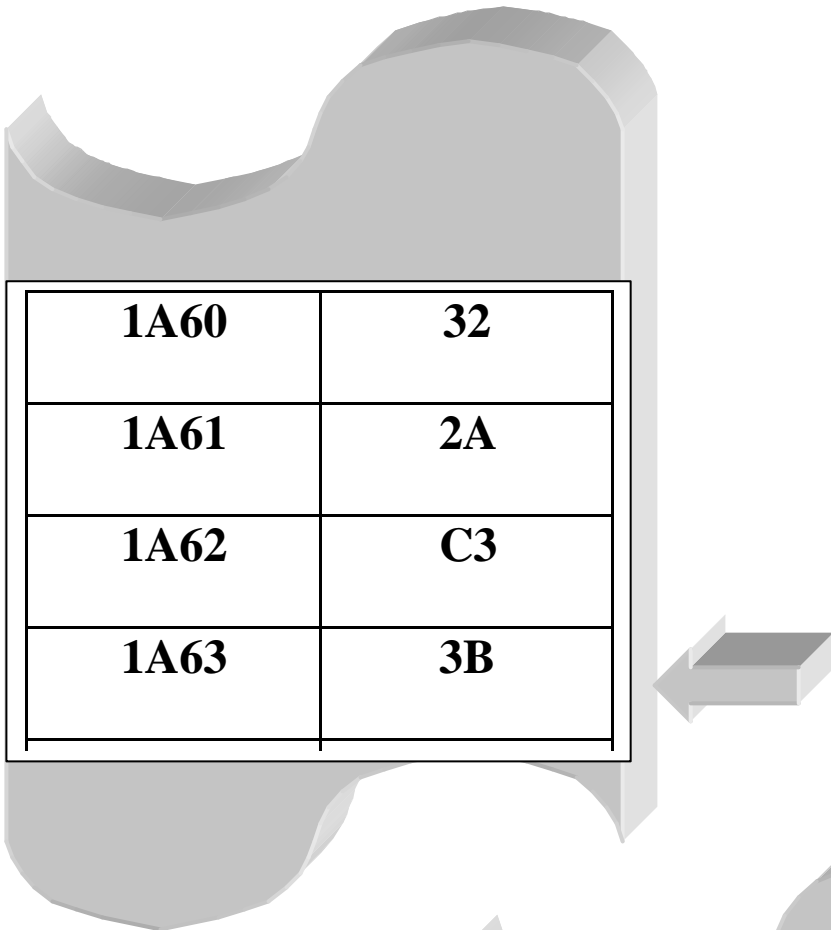
**0063**

**DE**

**1B62**

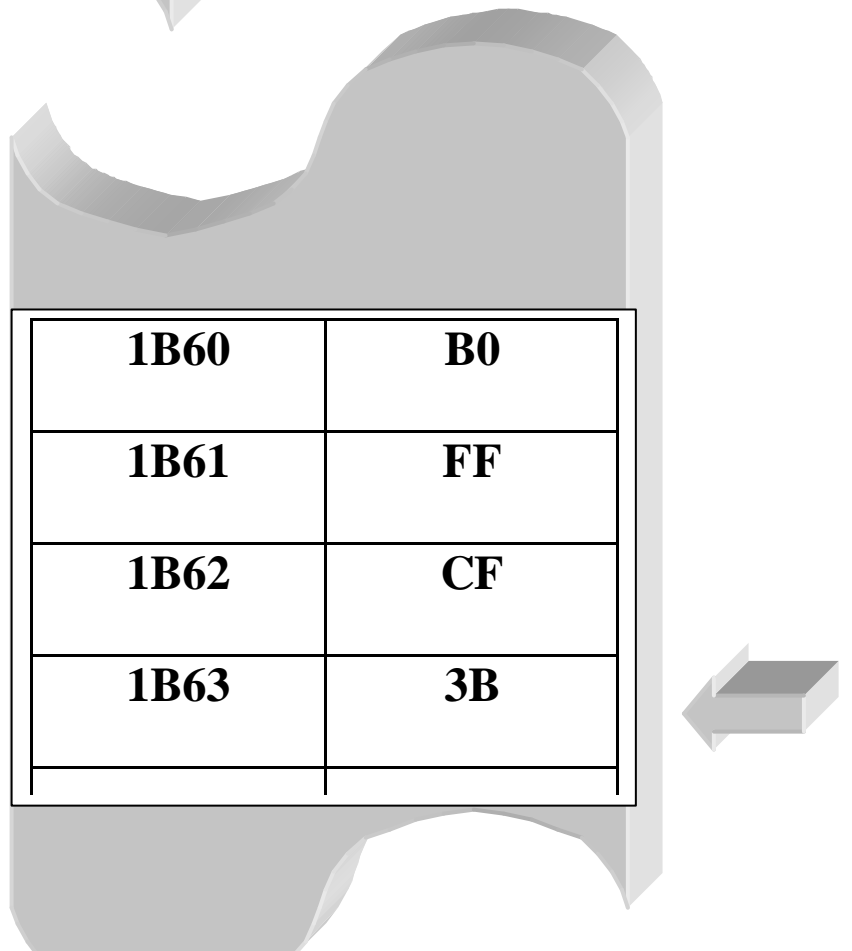
**HL**

**1A62**



A diagram showing a 4-bit field of a Z80 instruction. The field is represented as a vertical rectangle with a decorative top edge. Inside, a table lists four rows of bit patterns. A 3D arrow points to the right from the right side of the table.

<b>1A60</b>	<b>32</b>
<b>1A61</b>	<b>2A</b>
<b>1A62</b>	<b>C3</b>
<b>1A63</b>	<b>3B</b>



A diagram showing a 4-bit field of a Z80 instruction. The field is represented as a vertical rectangle with a decorative top edge. Inside, a table lists four rows of bit patterns. A 3D arrow points to the right from the right side of the table.

<b>1B60</b>	<b>B0</b>
<b>1B61</b>	<b>FF</b>
<b>1B62</b>	<b>CF</b>
<b>1B63</b>	<b>3B</b>

---

**Effetto sul registro dei flag**

<b>S</b>	<b>Z</b>	<b>H</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
		<b>0</b>	<b>X</b>	<b>0</b>	

- IL flag H ed il flag N sono posti a zero
- IL flag P/V viene utilizzato per sapere se si è completato il trasferimento dei dati, cioè se il contatore BC è arrivato a zero. Infatti se BC è diverso da zero dopo l'esecuzione dell'istruzione, il flag è posto ad uno, se BC è arrivato a zero , il flag viene posto a zero.

---

**LDDR**

Con l'istruzione LDD viene trasferito soltanto un dato. Per effettuare il trasferimento di tutto il blocco di dati occorre inserire l'istruzione in un ciclo, in modo da farne ripetere l'esecuzione fino a che il contatore BC non giunga a zero.

**Esempio**

**LD HL , 1A63 ; limite blocco sorgente**

**LD DE, 1B63 ; limite blocco destinazione**

**LD BC, 0004 ; caricamento contatore**

**ET: LDD**

**JP PE, ET ; PE =effettua il salto se P/V=1**

Alternativamente si può utilizzare l'istruzione LDDR. Quest'istruzione esegue tutte le funzioni dell'istruzione LDD (trasferimento di un dato dal blocco sorgente al blocco destinazione, decremento dei puntatori DE ed HI e del contatore BC) non una sola volta ma iterativamente fino a che il contatore non si sia azzerato.

**Esempio: trasferimento di un blocco di quattro byte; blocco sorgente dall'indirizzo 1A60h all'indirizzo 1A63h; blocco destinazione dall'indirizzo 1B60h all'indirizzo 1B63h**

**Prima**

**BC**

**0004**

**DE**

**1B63**

**HL**

**1A63**

<b>1A60</b>	<b>32</b>
<b>1A61</b>	<b>2A</b>
<b>1A62</b>	<b>C3</b>
<b>1A63</b>	<b>3B</b>

<b>1B60</b>	<b>B0</b>
<b>1B61</b>	<b>FF</b>
<b>1B62</b>	<b>CF</b>
<b>1B63</b>	<b>1F</b>

**Dopo**

**BC**

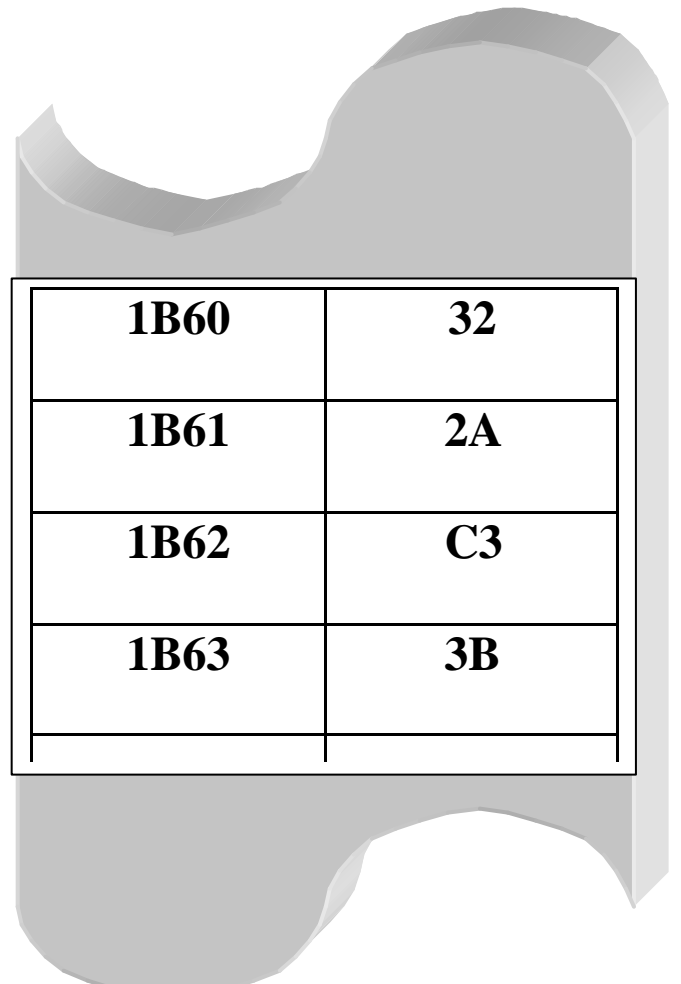
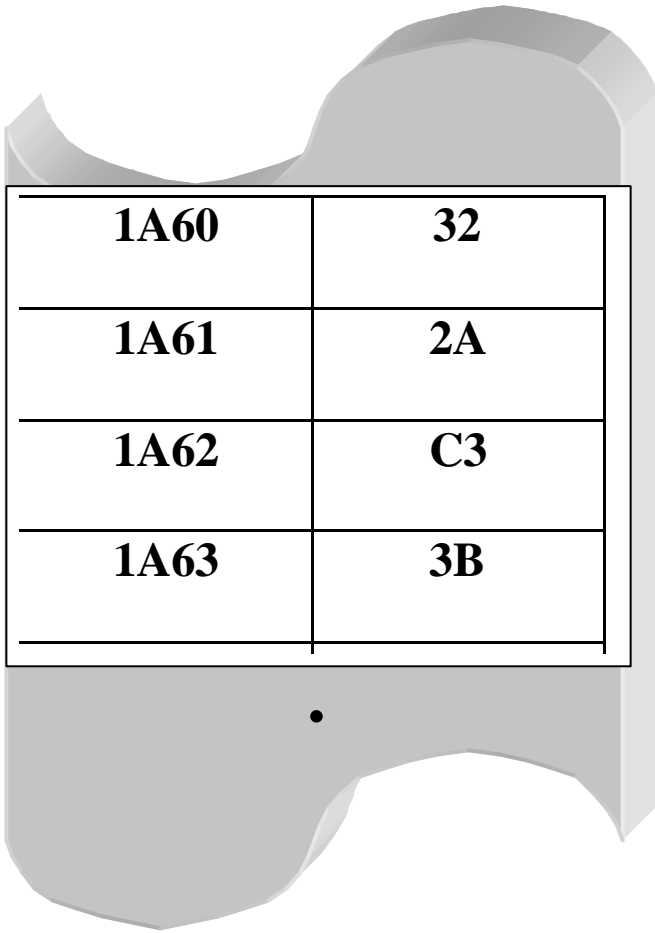
**0000**

**DE**

**1B5F**

**HL**

**1A5F**



**Effetto sul registro dei flag**

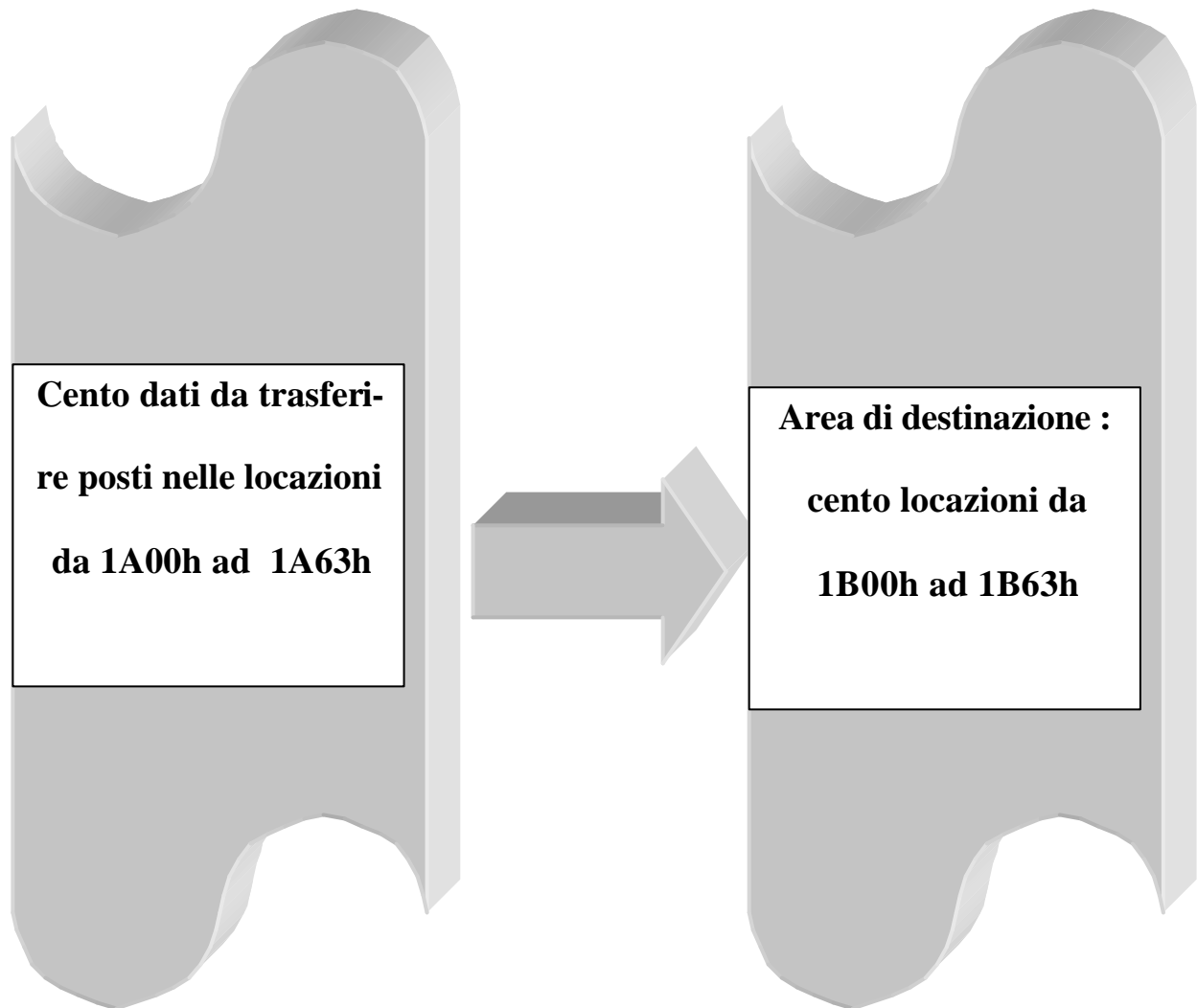
<b>S</b>	<b>Z</b>		<b>H</b>		<b>P/V</b>	<b>N</b>	<b>C</b>
			<b>0</b>		<b>0</b>	<b>0</b>	

- IL flag H, il flag PV ed il flag N sono posti a zero

---

**LDI**

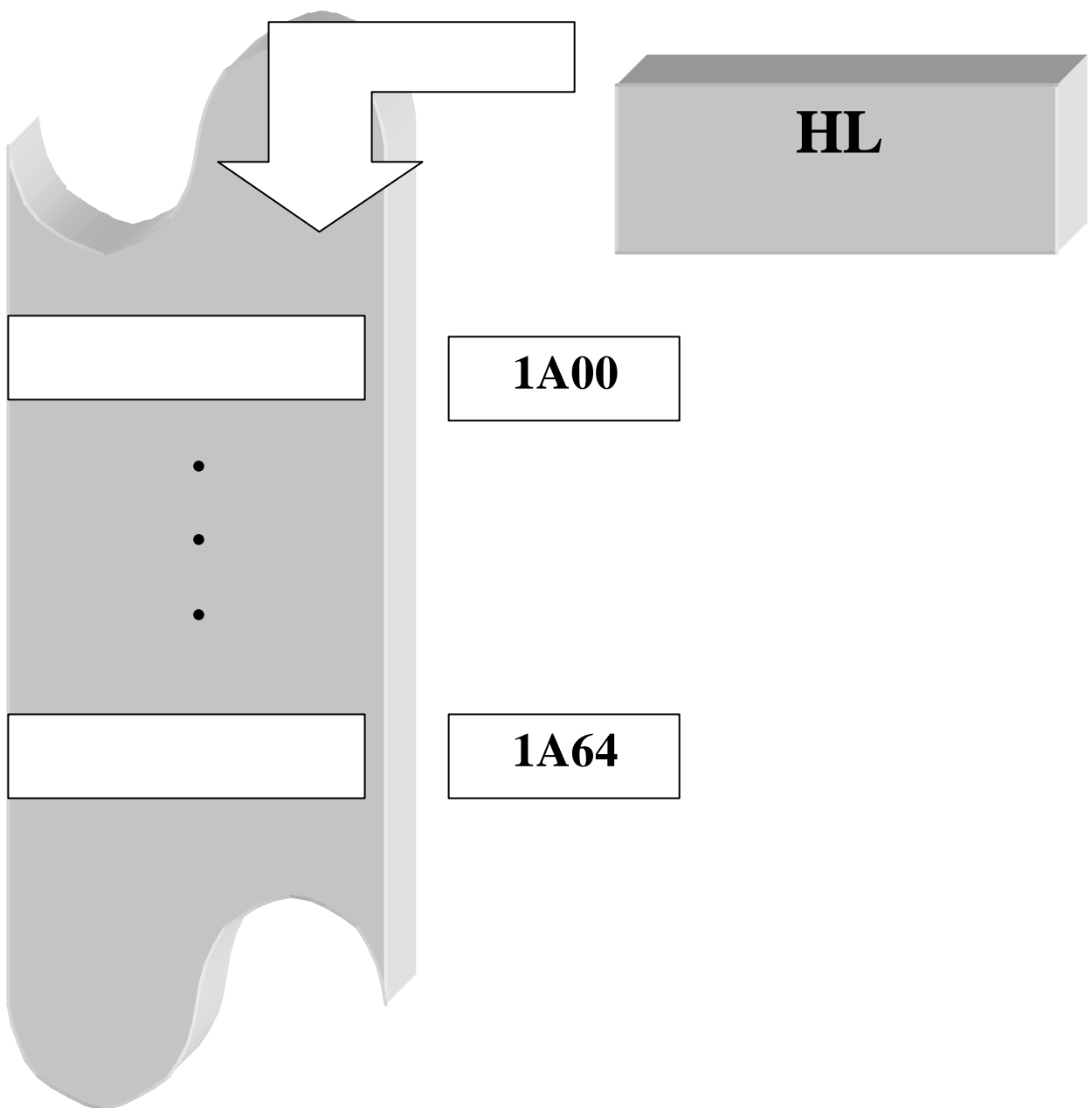
Questa istruzione viene utilizzata quando si vuole trasferire un blocco di dati da una zona all'altra della memoria

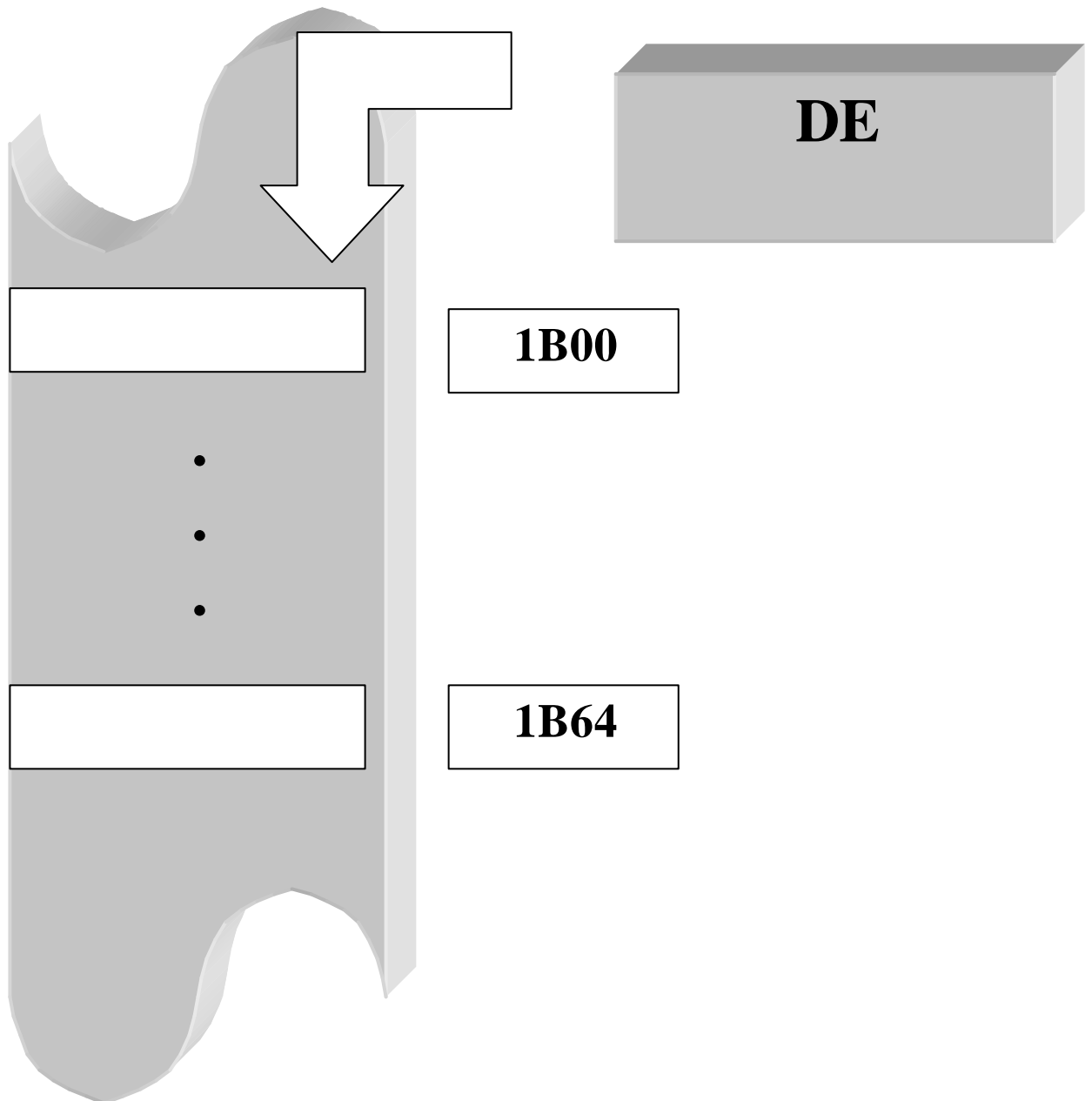


In questo caso occorrono

- un puntatore alla zona di memoria sorgente dei dati, cioè un registro a 16 bit contenente l'indirizzo della locazione da cui prelevare il dato che è il registro HL,

- un puntatore alla zona di memoria destinazione dei dati, cioè un registro a 16 bit contenente l'indirizzo della locazione in cui spostare il dato, che è il registro DE
- un registro che faccia da contatore per conoscere quando sarà stato terminato il processo di trasferimento dei dati, che è il registro BC





In questa istruzione si parte dalle locazioni di indirizzo più basso e si prosegue verso le locazioni di indirizzo più elevato.

Ogni volta che viene eseguita l'istruzione viene letto il dato contenuto nella locazione puntata da HL, viene trasferito nella locazione puntata da DE, vengono incrementati HL, DE e decrementato il contatore BC.

**(DE) ← (HL)**

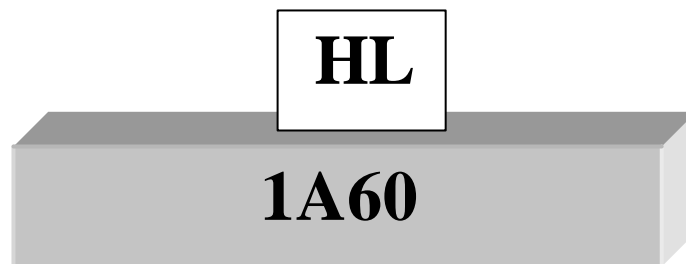
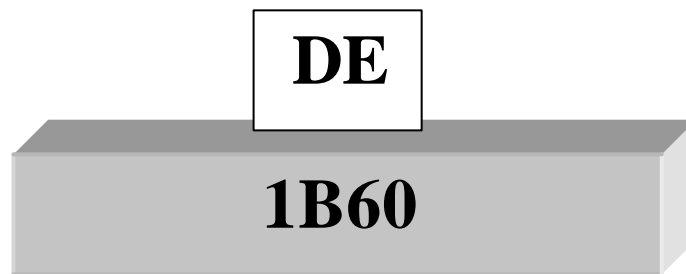
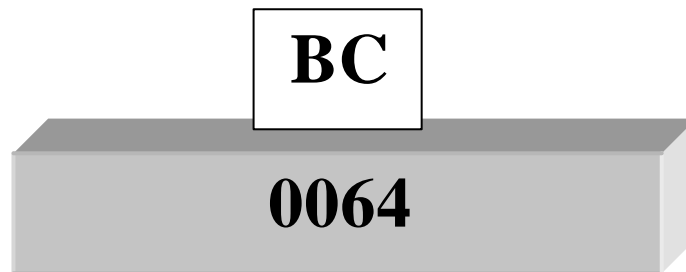
**DE=DE+1**

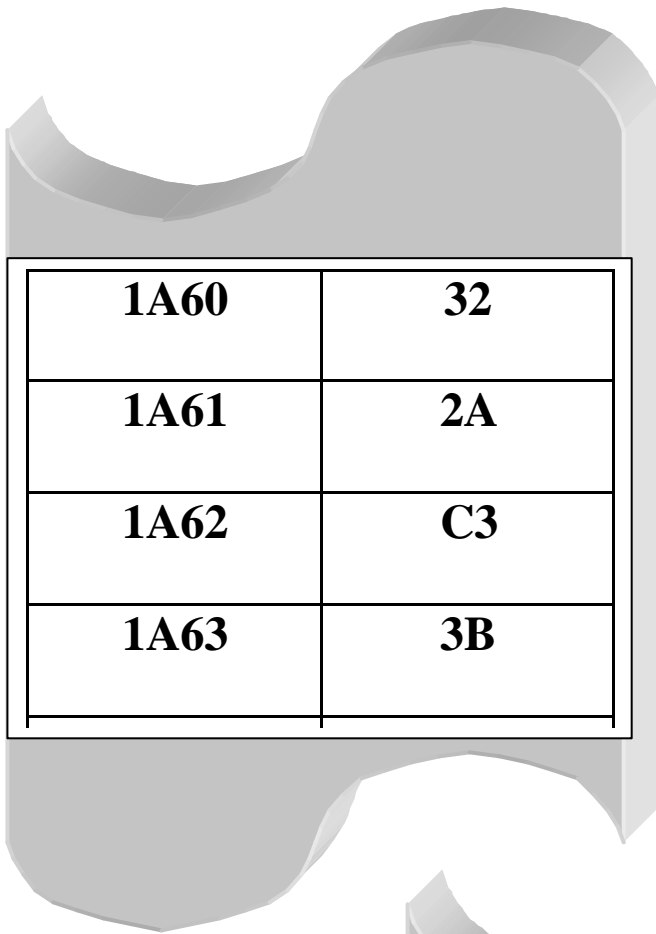
**HL=HL+1**

**BC=BC-1**

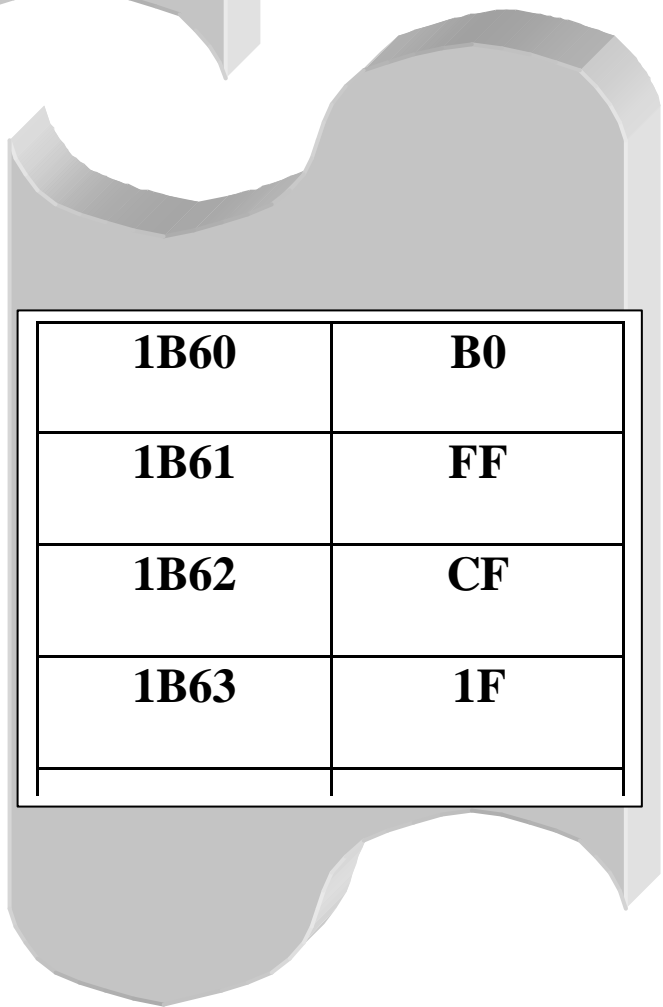
**Esempio**

**Prima**





<b>1A60</b>	<b>32</b>
<b>1A61</b>	<b>2A</b>
<b>1A62</b>	<b>C3</b>
<b>1A63</b>	<b>3B</b>



<b>1B60</b>	<b>B0</b>
<b>1B61</b>	<b>FF</b>
<b>1B62</b>	<b>CF</b>
<b>1B63</b>	<b>1F</b>

**Dopo**

**BC**

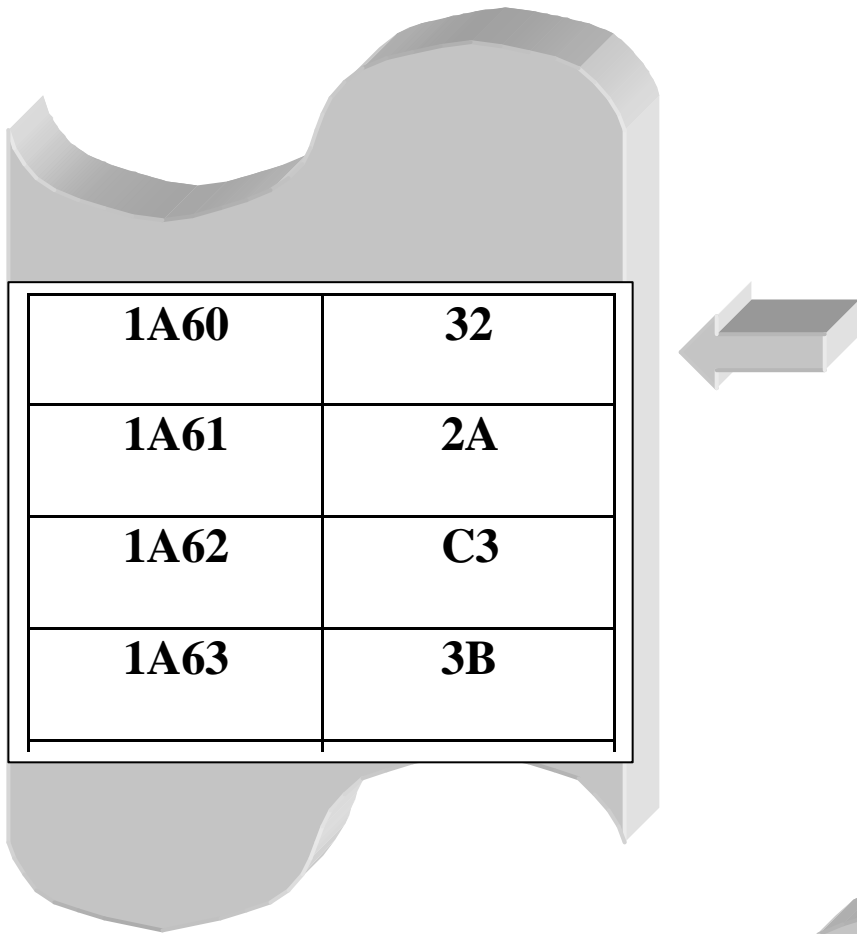
**0063**

**DE**

**1B61**

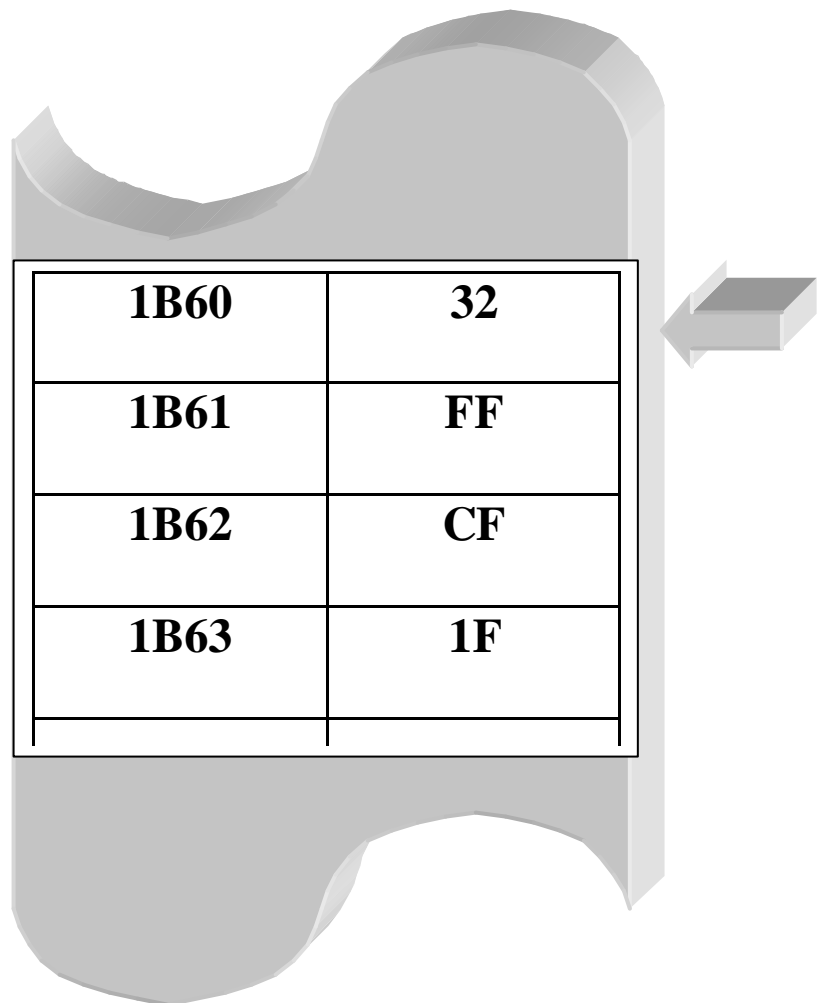
**HL**

**1A61**



A diagram showing a Z80 instruction format. A grey 3D-style shape represents the instruction, with a white rectangular box overlaid on its left side. This box contains a table with four rows and two columns. A grey arrow points from the right side of the table towards the instruction shape.

<b>1A60</b>	<b>32</b>
<b>1A61</b>	<b>2A</b>
<b>1A62</b>	<b>C3</b>
<b>1A63</b>	<b>3B</b>



A diagram showing a Z80 instruction format. A grey 3D-style shape represents the instruction, with a white rectangular box overlaid on its left side. This box contains a table with four rows and two columns. A grey arrow points from the right side of the table towards the instruction shape.

<b>1B60</b>	<b>32</b>
<b>1B61</b>	<b>FF</b>
<b>1B62</b>	<b>CF</b>
<b>1B63</b>	<b>1F</b>

---

**Effetto sul registro dei flag**

<b>S</b>	<b>Z</b>	<b>H</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
		<b>0</b>	<b>X</b>	<b>0</b>	

- IL flag H ed il flag N sono posti a zero
- IL flag P/V viene utilizzato per sapere se si è completato il trasferimento dei dati, cioè se il contatore BC è arrivato a zero. Infatti se BC è diverso da zero dopo l'esecuzione dell'istruzione, il flag è posto ad uno, se BC è arrivato a zero , il flag viene posto a zero.

---

**LDIR**

Con l'istruzione LDI viene trasferito soltanto un dato. Per effettuare il trasferimento di tutto il blocco di dati occorre inserire l'istruzione in un ciclo, in modo da farne ripetere l'esecuzione fino a che il contatore BC non giunga a zero.

**Esempio**

**LD HL , 1A60 ; limite inferiore blocco sorgente**

**LD DE, 1B60 ; limite blocco destinazione**

**LD BC, 0004 ; caricamento contatore**

**ET: LDI**

**JP PE, ET ; PE =effettua il salto se P/V=1**

Alternativamente si può utilizzare l'istruzione LDIR. Quest'istruzione esegue tutte le funzioni dell'istruzione LDD (trasferimento di un dato dal blocco sorgente al blocco destinazione, incremento dei puntatori DE ed HI e decremento del contatore BC) non una sola volta ma iterativamente fino a che il contatore non si sia azzerato.

**Esempio: trasferimento di un blocco di quattro byte; blocco sorgente dall'indirizzo 1A60h all'indirizzo 1A63h; blocco destinazione dall'indirizzo 1B60h all'indirizzo 1B63h**

**Prima**

**BC**

**0004**

**DE**

**1B60**

**HL**

**1A60**

<b>1A60</b>	<b>32</b>
<b>1A61</b>	<b>2A</b>
<b>1A62</b>	<b>C3</b>
<b>1A63</b>	<b>3B</b>

<b>1B60</b>	<b>B0</b>
<b>1B61</b>	<b>FF</b>
<b>1B62</b>	<b>CF</b>
<b>1B63</b>	<b>1F</b>

**Dopo**

**BC**

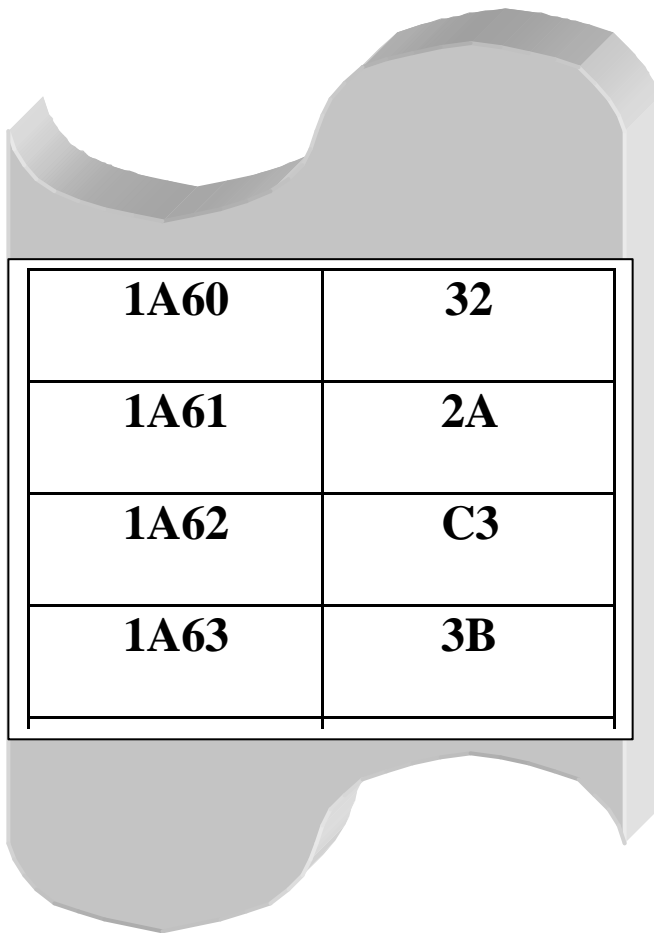
**0000**

**DE**

**1B64**

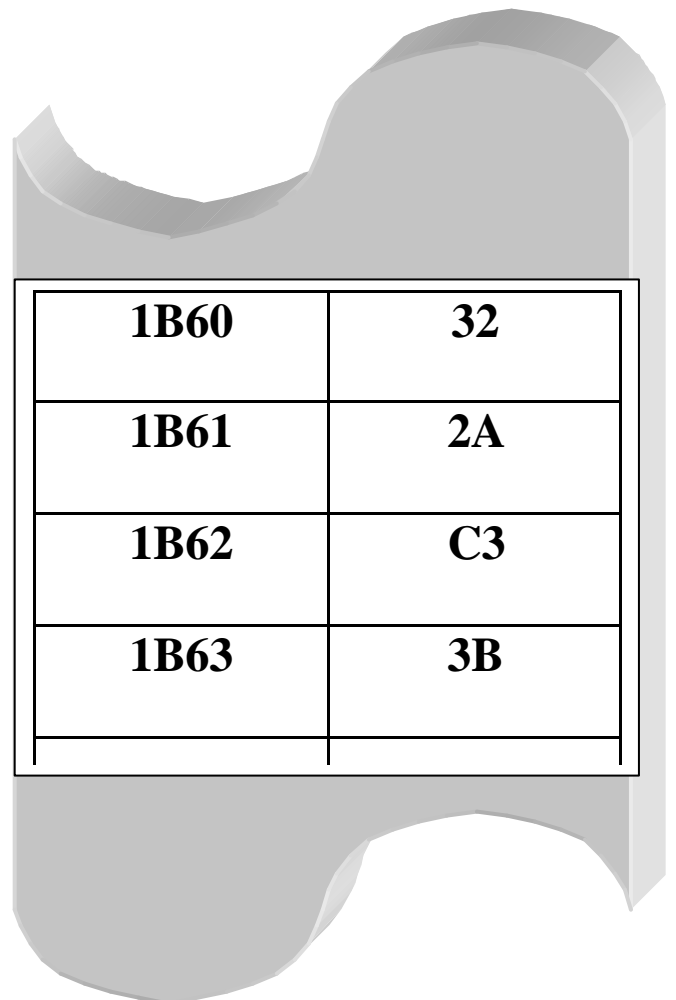
**HL**

**1A64**



A diagram of a Z80 instruction format, represented as a grey, irregular shape with a central white rectangular area containing a table. The table lists four bit fields: 1A60 (32 bits), 1A61 (2A), 1A62 (C3), and 1A63 (3B).

<b>1A60</b>	<b>32</b>
<b>1A61</b>	<b>2A</b>
<b>1A62</b>	<b>C3</b>
<b>1A63</b>	<b>3B</b>



A diagram of a Z80 instruction format, represented as a grey, irregular shape with a central white rectangular area containing a table. The table lists four bit fields: 1B60 (32 bits), 1B61 (2A), 1B62 (C3), and 1B63 (3B).

<b>1B60</b>	<b>32</b>
<b>1B61</b>	<b>2A</b>
<b>1B62</b>	<b>C3</b>
<b>1B63</b>	<b>3B</b>

---

**Effetto sul registro dei flag**

<b>S</b>	<b>Z</b>	<b>H</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>0</b>	

- IL flag H, il flag PV ed il flag N sono posti a zero

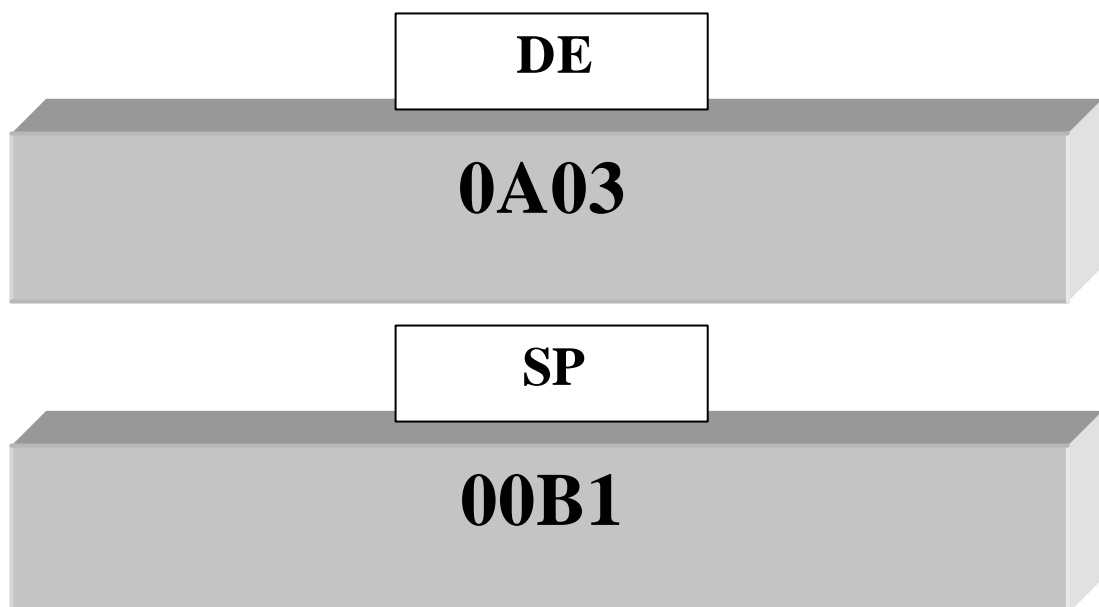
---

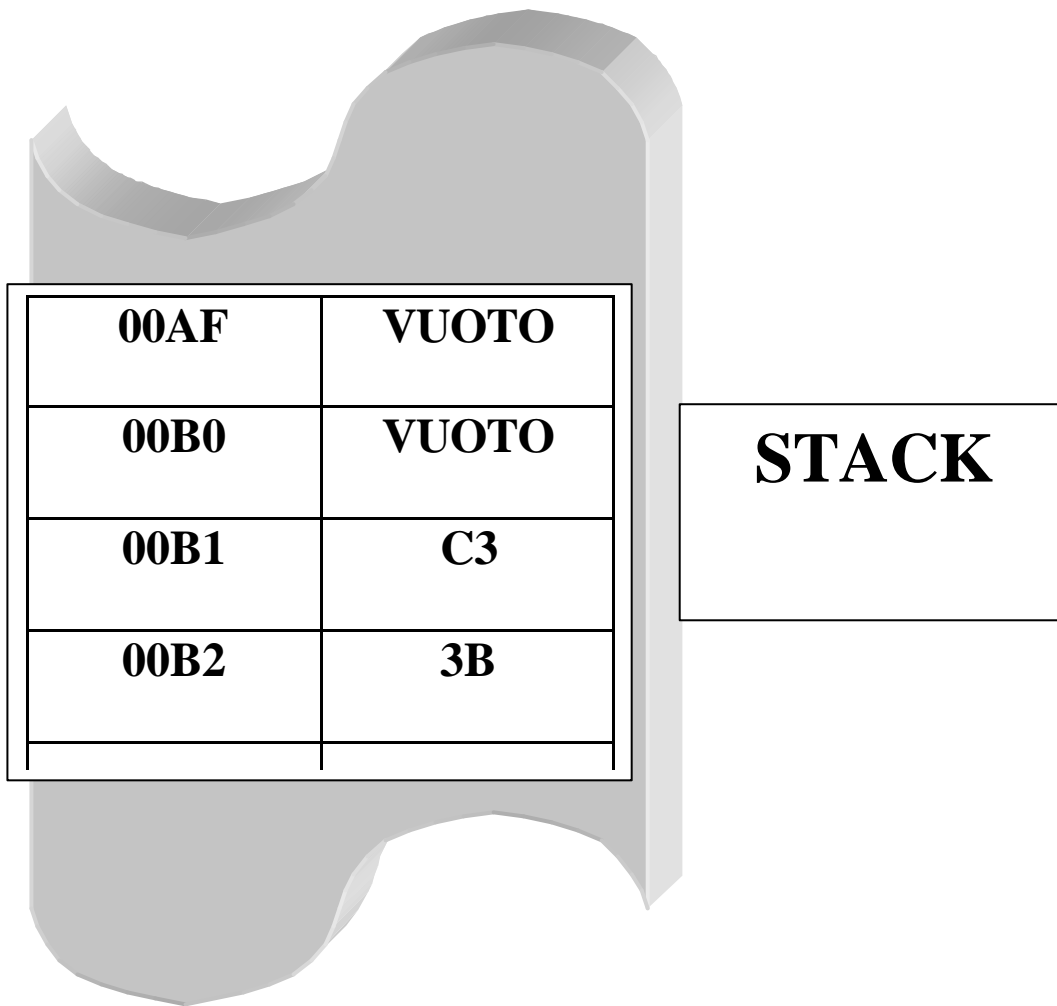
**PUSH qq**

Con quest'istruzione la coppia di registri o registro a 16 bit qq viene spinta in cima allo stack. Ricordiamo che lo Stack Pointer punta all'ultima locazione riempita dello stack e che questo cresce occupando le locazioni di memoria che hanno indirizzo via via decrescente.

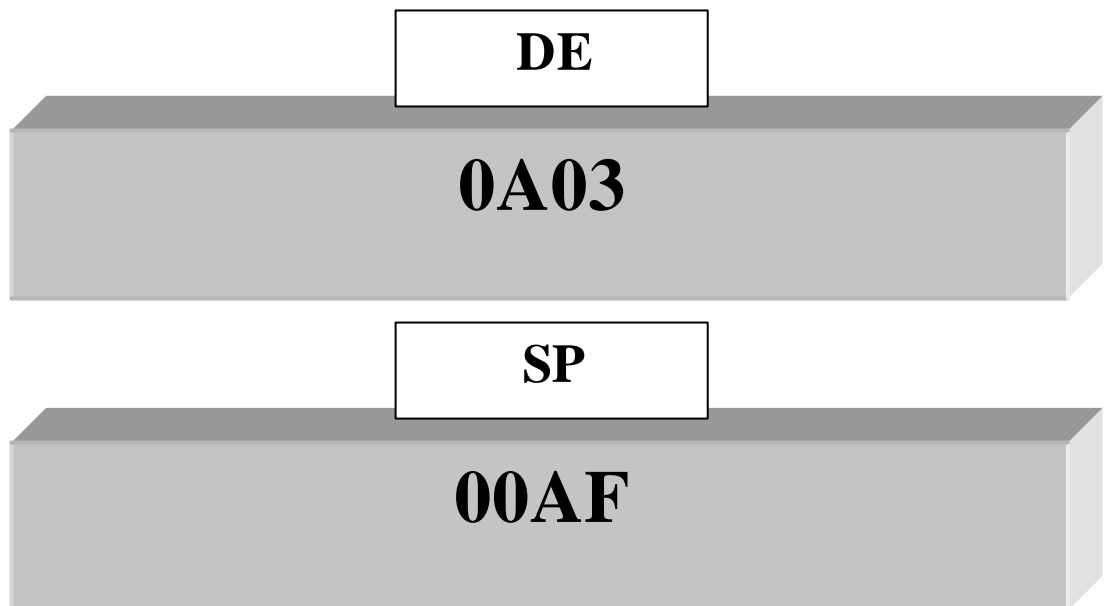
***Allora la parte alta del registro qq va nella prima locazione libera, quindi nella locazione il cui indirizzo si ottiene dall'indirizzo contenuto in SP meno uno, mentre la parte bassa di qq va nella locazione successiva, quindi nella locazione il cui indirizzo si ottiene dall'indirizzo contenuto in SP meno due. Poiché lo stack cresce di due locazioni, lo Stack Pointer va decrementato di due unità***

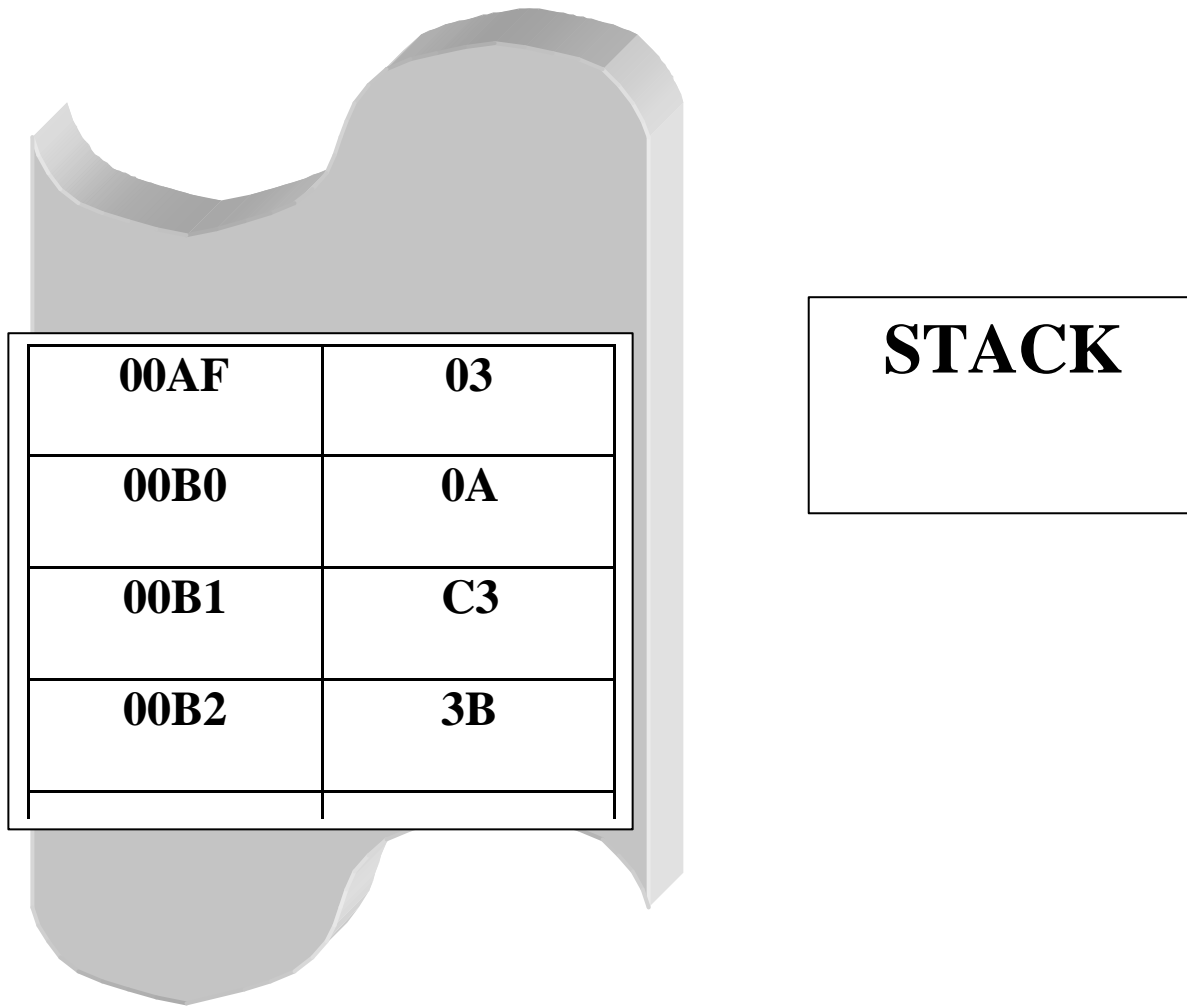
Non si ha alcun effetto sul registro dei flag.

**Esempio: PUSH DE****Prima**



**Dopo**





qq può essere uno dei seguenti registri: BC; DE; HL; AF; IX; IY

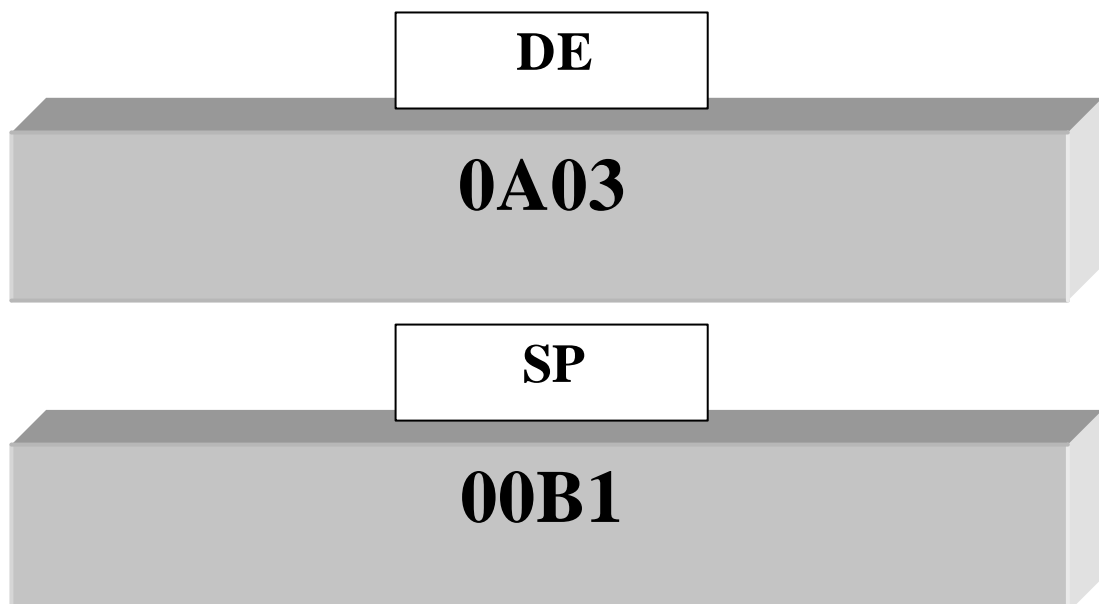
---

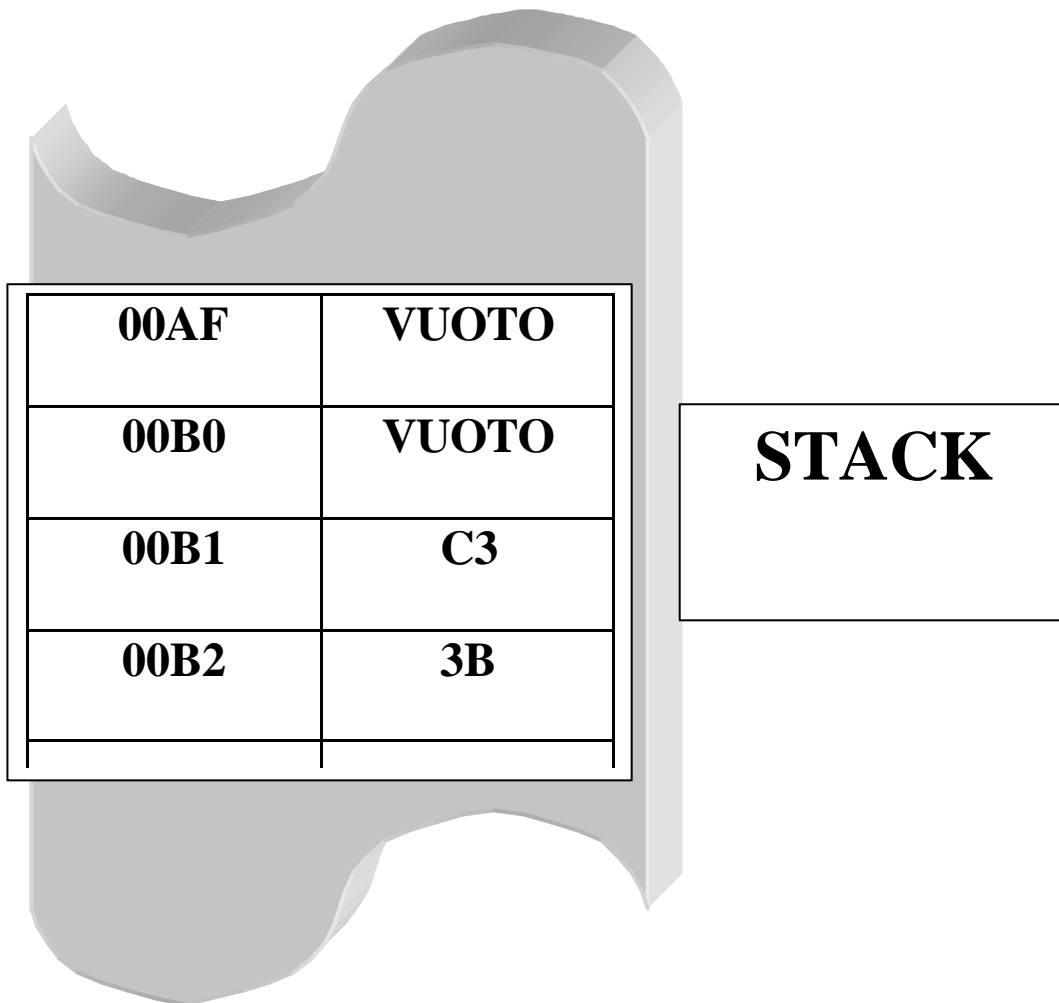
**POP qq**

Con quest'istruzione la coppia di registri o registro a 16 bit qq viene riempita con il contenuto delle due locazioni di memoria in cima allo stack. Ricordiamo che lo Stack Pointer punta all'ultima locazione riempita dello stack e che questo cresce occupando le locazioni di memoria che hanno indirizzo via via decrescente.

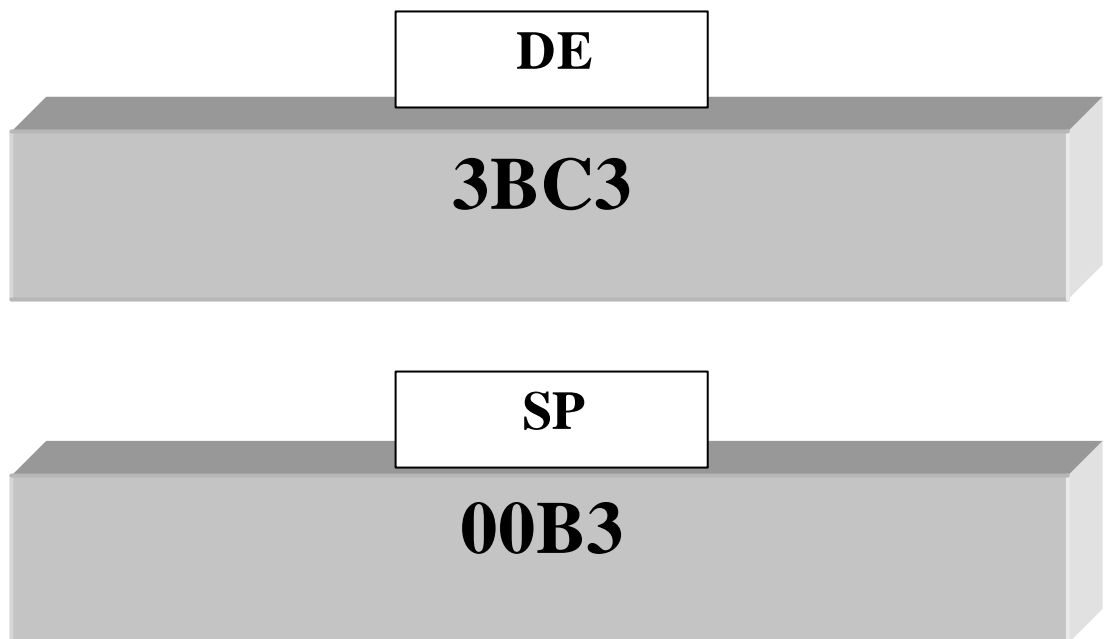
***Allora la parte alta del registro qq viene riempita con il contenuto della penultima locazione occupata, quindi nella locazione il cui indirizzo si ottiene dall'indirizzo contenuto in SP più uno, mentre la parte bassa di qq viene riempita con il contenuto della locazione successiva, quindi nella locazione il cui indirizzo si ottiene dall'indirizzo contenuto in SP. Poiché lo stack decresce di due locazioni, lo Stack Pointer va incrementato di due unità***

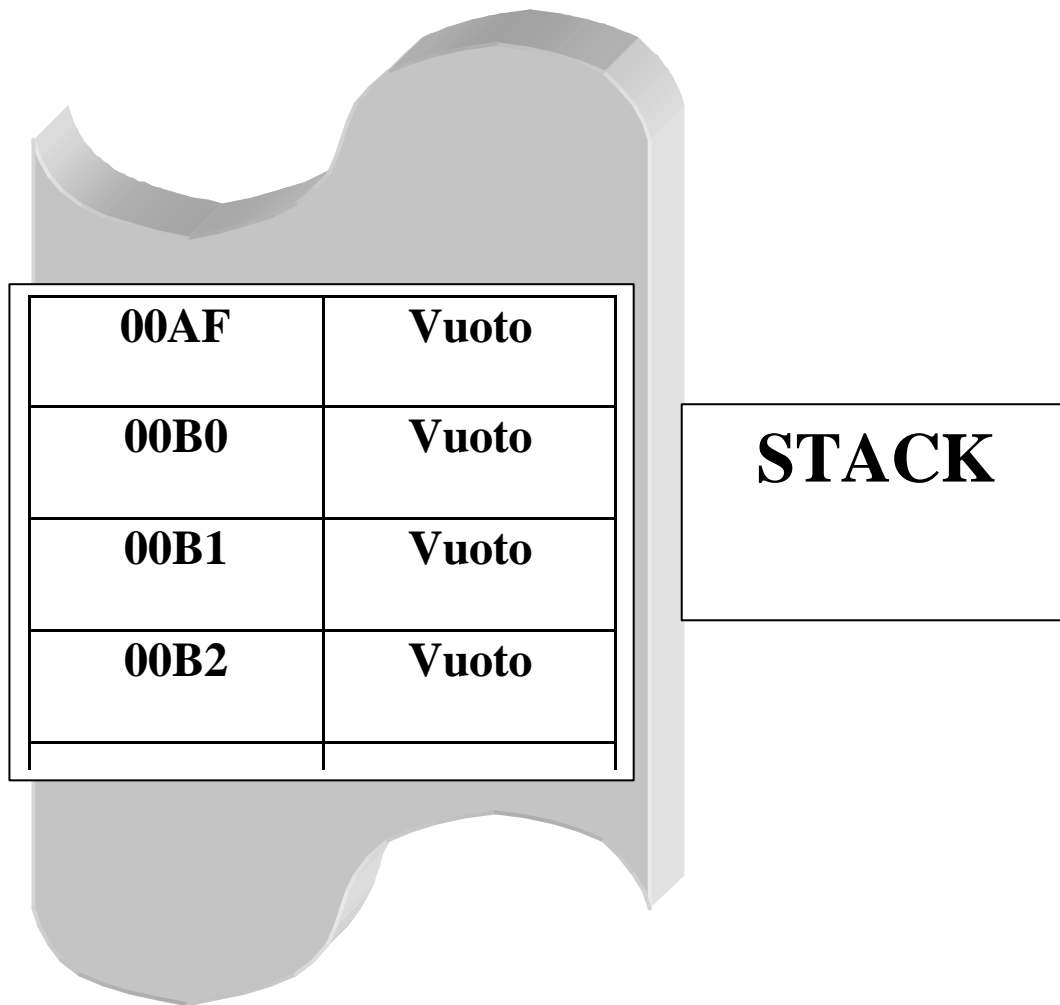
Non si ha alcun effetto sul registro dei flag.

**Esempio: POP DE****Prima**



**Dopo**



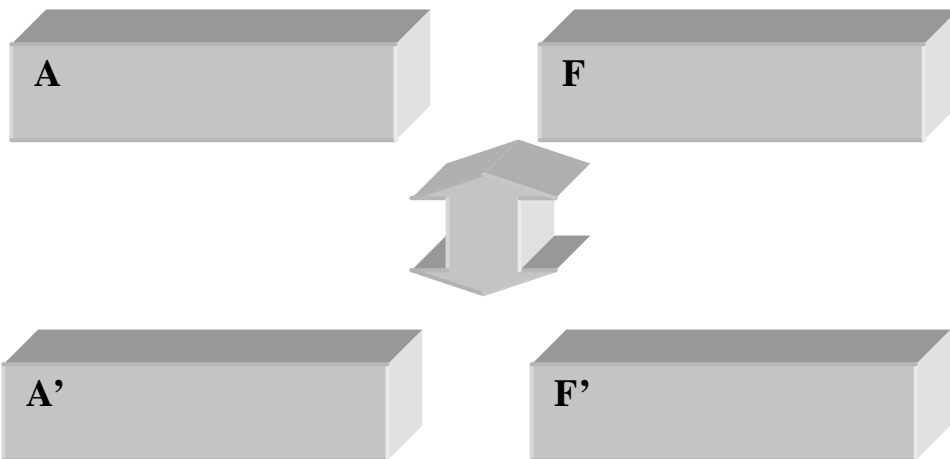


qq può essere uno dei seguenti registri: BC; DE; HL; AF; IX; IY

*Istruzioni di scam-  
bio  
di dati*

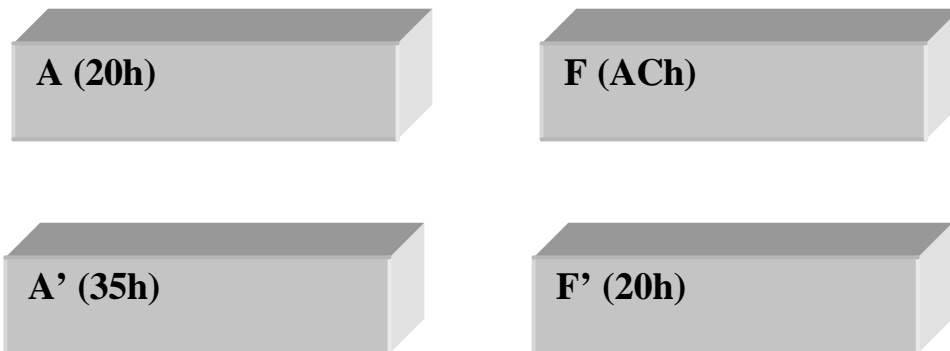
**EX AF, AF'**

Λειτουργία ανταλλαγής περιεχομένου των registrov A ed Φ con A' ed Φ'

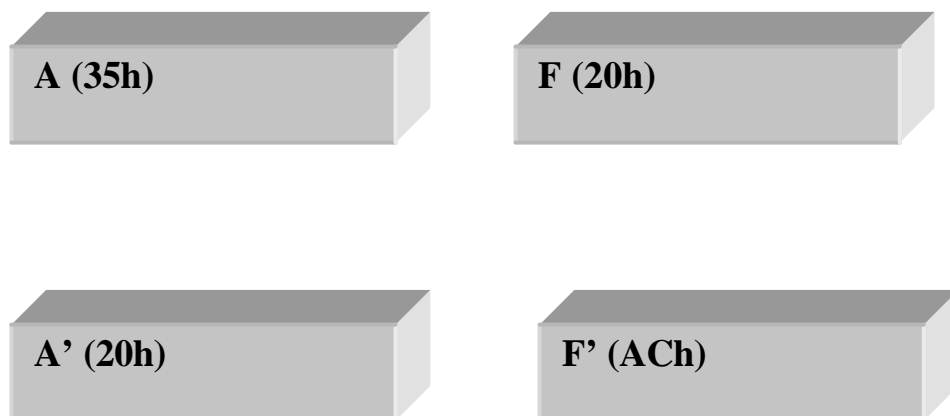


Εσεμπιο:

Χοντενυτο dei registrov prima dell'operazione di scambio EE



Χοντενυτο dei registrov dopo l'operazione di scambio EE



---

**EXX**

Λειτουργιόνε ΕΞΕ χονσεντε δι σχαμβιαρε ιλ χοντενυτο δει ρεγιστρι ΒΧ, ΔΕ, ΗΛ ε δει ρεγιστρι ΒΧε, ΔΕε, ΗΛε.

Εσεμπιο.

Πριμα



Δοπο

**B (2Ah)**

**C (F0h)**

**D (AFh)**

**E (B0h)**

**H (25h)**

**L (3Bh)**

**B' (20h)**

**C' (10h)**

**D' (A0h)**

**E' (B3h)**

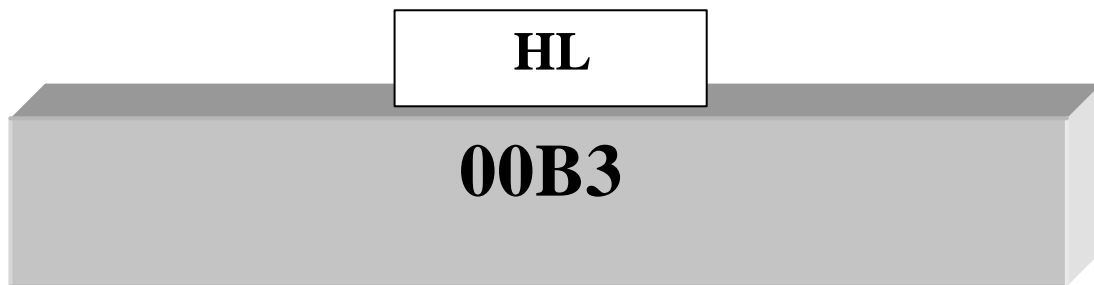
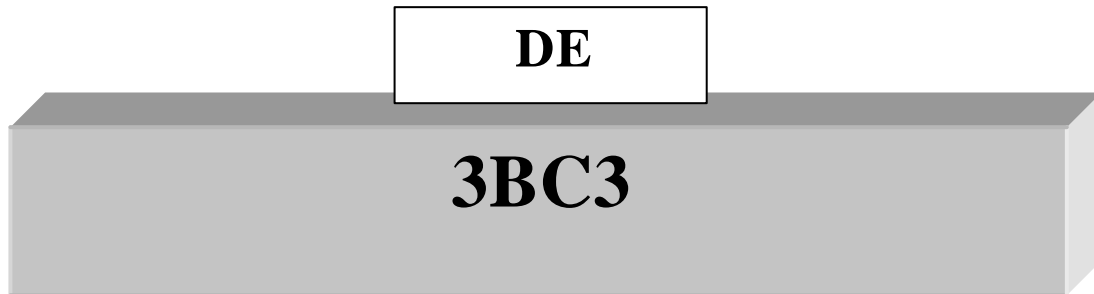
**H' (45h)**

**L' (30h)**

---

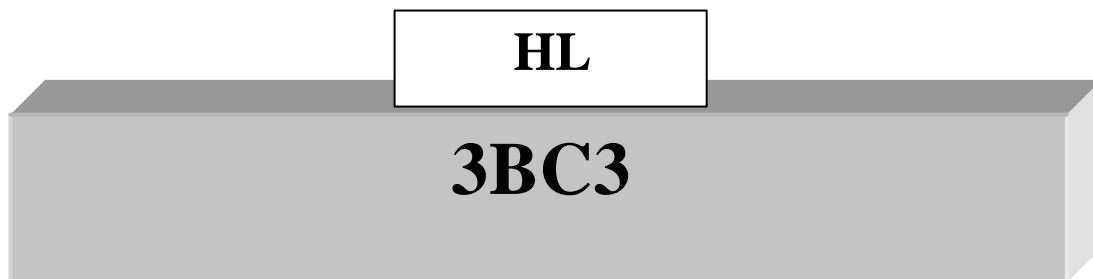
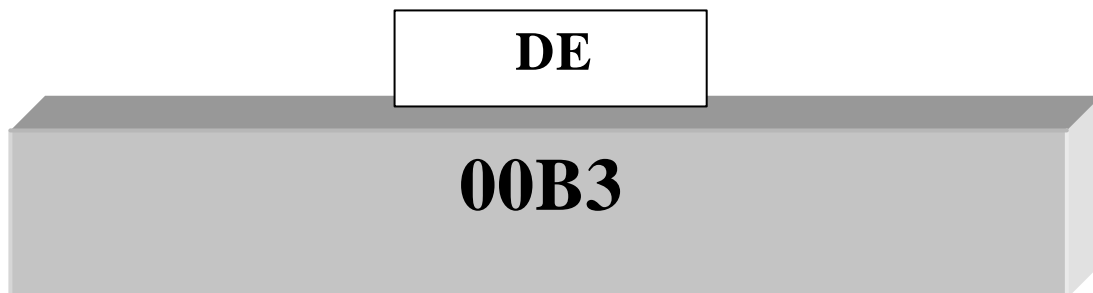
**EX DE, HL**

Questa istruzione scambia il contenuto dei registri DE ed HL



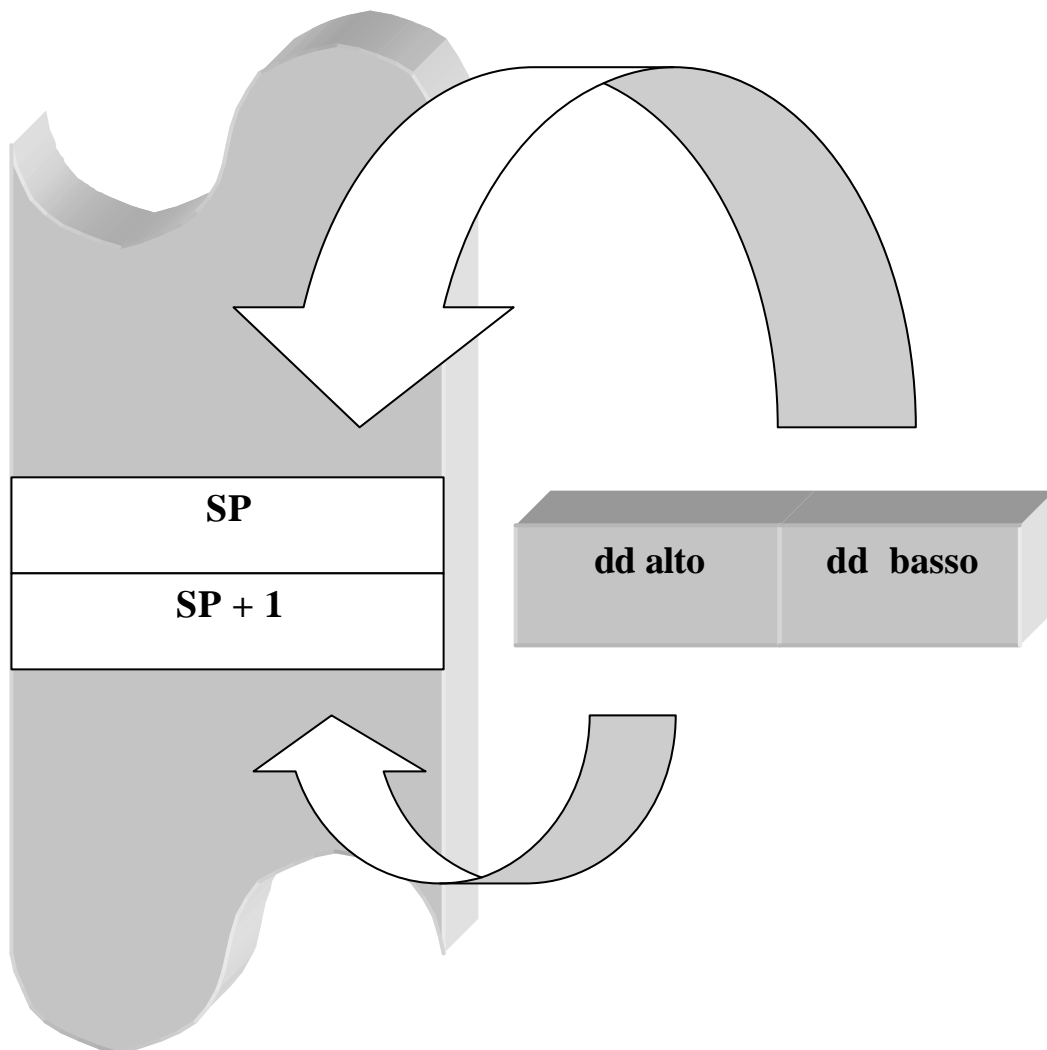
**Esempio Prima**

**Dopo**



**EX (SP), dd**

Il contenuto del registro a 16 bit dd viene scambiato con le due locazioni di memoria che stanno in cima allo stack



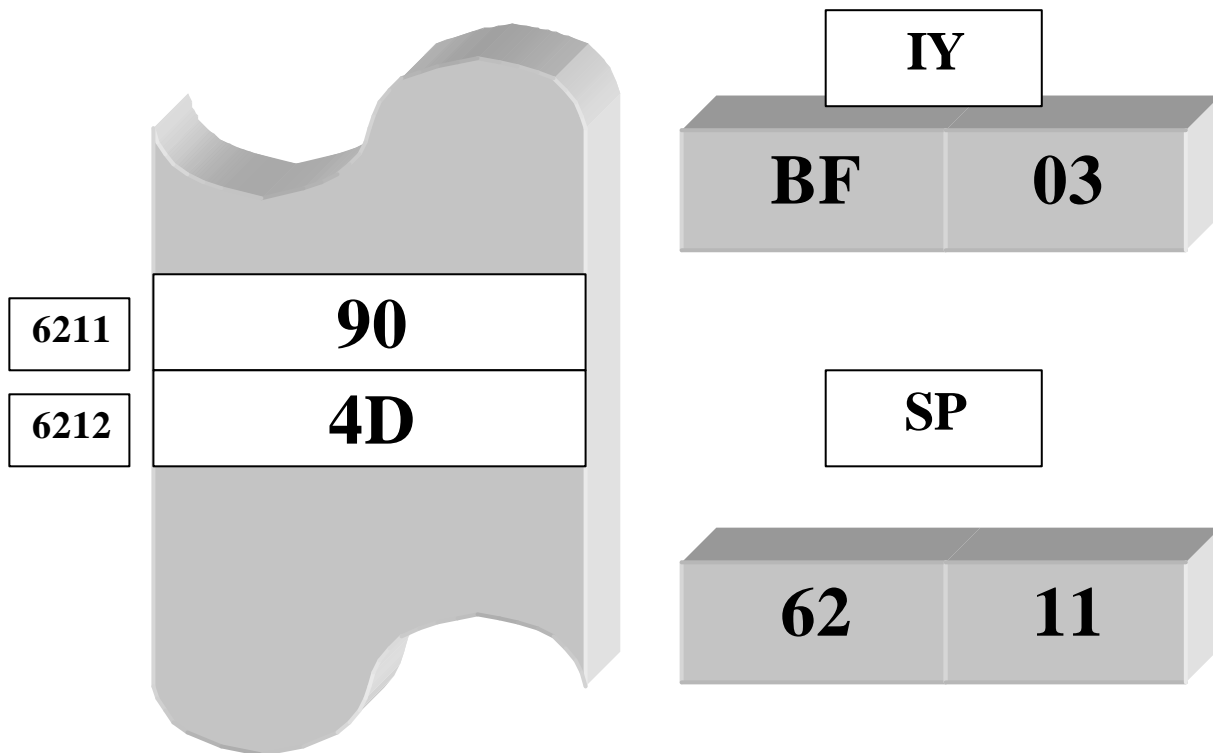
La parte

bassa del registro viene scambiata con la locazione puntata dallo SP, la parte alta del registro viene scambiata con la locazione il cui indirizzo è dato da SP+1. Il registro dd può essere: HL, IX, IY. Poiché l'ampiezza dello stack non viene modificata in quanto non si aggiungono dati nelle locazioni vuote, né si prelevano dati svuotando locazioni, lo SP rimane inalterato.

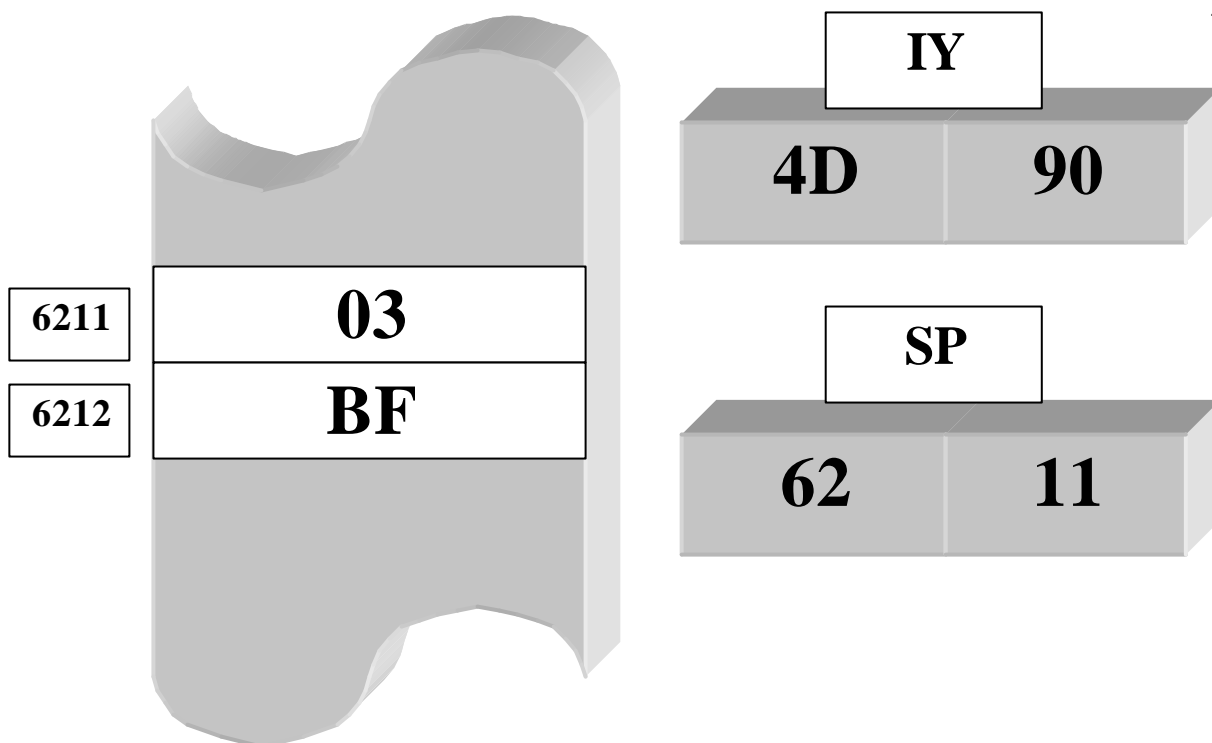
*L'istruzione non ha alcun effetto sul registro dei flag.*

**Esempio: EX (SP), IY**

**Prima**



**Dopo**



*Istruzioni di ricer-*  
*ca*

---

**CP A, s**

Con quest'istruzione viene confrontato il contenuto dell'accumulatore con l'operando s. L'istruzione serve, ad esempio, se si vuole ricercare un dato in una zona di memoria, allora si pone nell'accumulatore il dato da cercare mentre come operando s considereremo le varie locazioni della zona di memoria in cui stiamo effettuando la ricerca. Se l'operazione di confronto da un esito positivo vuol dire che vi è una locazione il cui contenuto coincide con quello dell'accumulatore, cioè in quella locazione esiste il dato cercato.

L'operazione di confronto viene effettuata dallo Z80 con un'operazione di sottrazione di cui, però, non viene memorizzato il risultato. La sottrazione viene effettuato soltanto perché in questomodo verrà modificato il registro dei flag. Se il flag Z sarà settato ad 1 vorrà dire che la sottrazione avrà dato un risultato nullo e quindi il contenuto dell'accumulatore sarà identico al contenuto dell'operando s. Controllando il flag S potremo invece sapere se il contenuto dell'accumulatore è maggiore o minore di quello dell'operando s. Infatti, se  $S = 0$ , la sottrazione ha dato un risultato positivo per cui il contenuto dell'accumulatore è maggiore di quello dell'operando s, se  $S = 1$  l'operazione ha dato risultato negativo il che vuol dire che il contenuto dell'accumulatore è minore di quello dell'operando s.

L'operando s può essere uno dei seguenti:

---

<b>CP r</b>	Indirizzamento a registro	Si sottrae all'accumulatore il contenuto del registro ad otto bit (A, B, C, D, E, H, L)
<b>CP n</b>	Indirizzamento immediato	Si sottrae all'accumulatore il contenuto del dato ad otto bit immediatamente espresso nell'istruzione
<b>CP (HL)</b>	Indirizzamento indiretto	Si sottrae all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro HL
<b>CP (IX+d)</b>	Indirizzamento indizzato	Si sottrae all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IX sommato allo spiazzamento d espresso nell'istruzione
<b>CP (IY+d)</b>	Indirizzamento indizzato	Si sottrae all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal con-

---

tenuto del registro IY sommato  
allo spiazzamento d espresso  
nell'istruzione

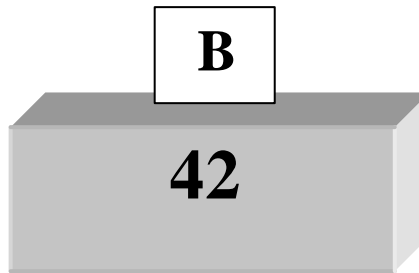
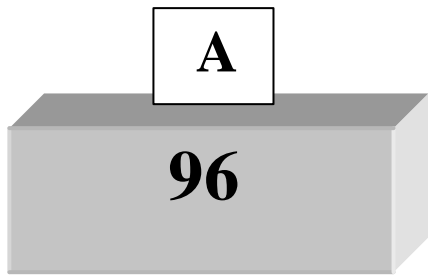
**Modifiche al registro dei flag****S****Z****H****P/V****N****C**

●	●		●		●	<b>1</b>	●
---	---	--	---	--	---	----------	---

IL flag N è posto sempre ad uno, gli altri flag dipendono dal risultato della sottrazione.

**Esempio:**

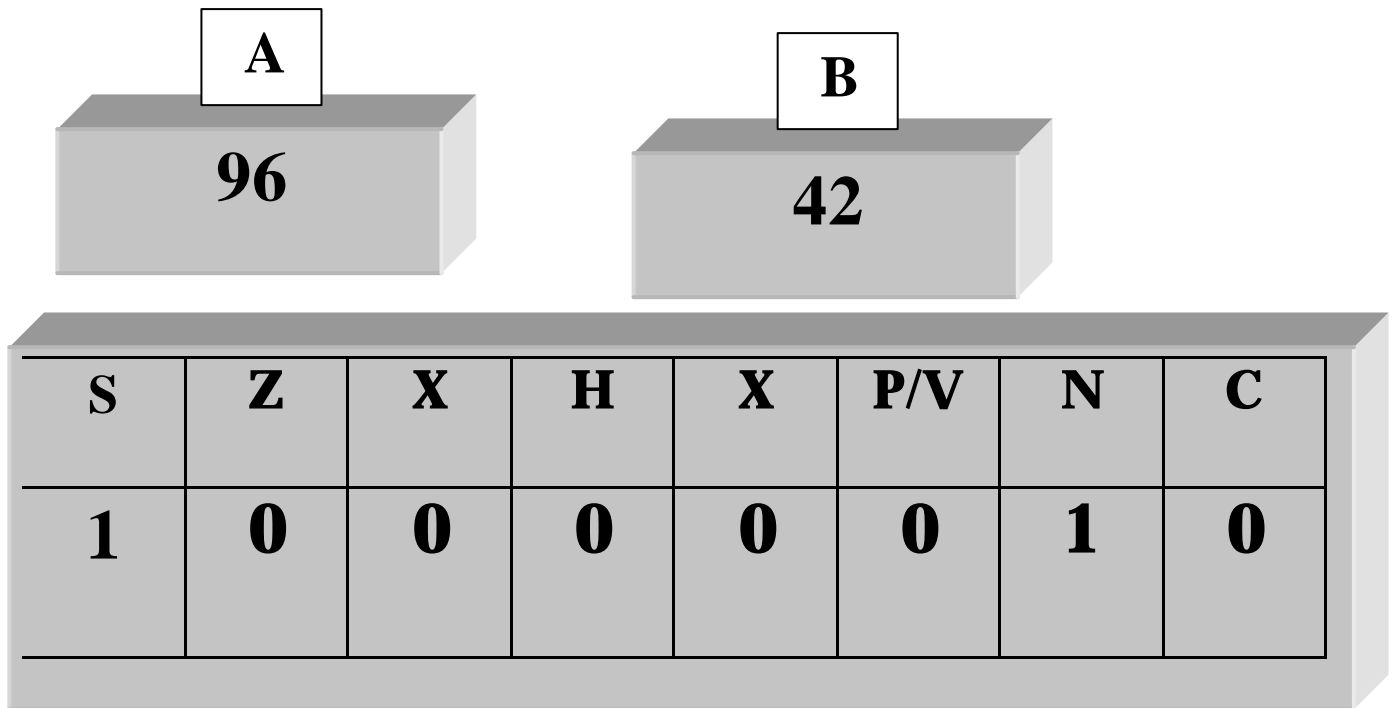
**CP B**



<b>S</b>	<b>Z</b>	<b>X</b>	<b>H</b>	<b>X</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>

**Prima**

**Dopo**



Per capire l'esempio ricordiamo che

$$96|_{16} = 10010110|_2$$

quindi rappresenta un numero negativo essendo l'MSB pari ad 1 mentre

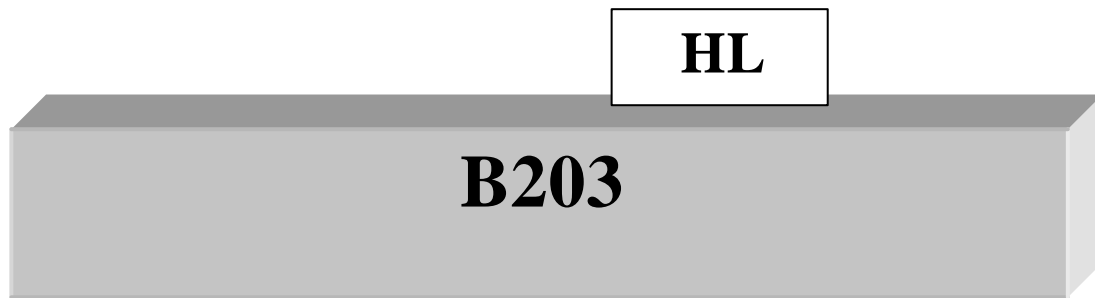
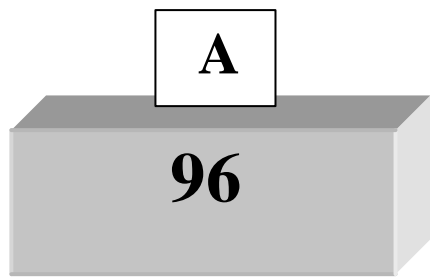
$$42|_{16} = 01000010|_2$$

per cui il dato contenuto nell'accumulatore è minore di quello contenuto nel registro B.

**Esempio**

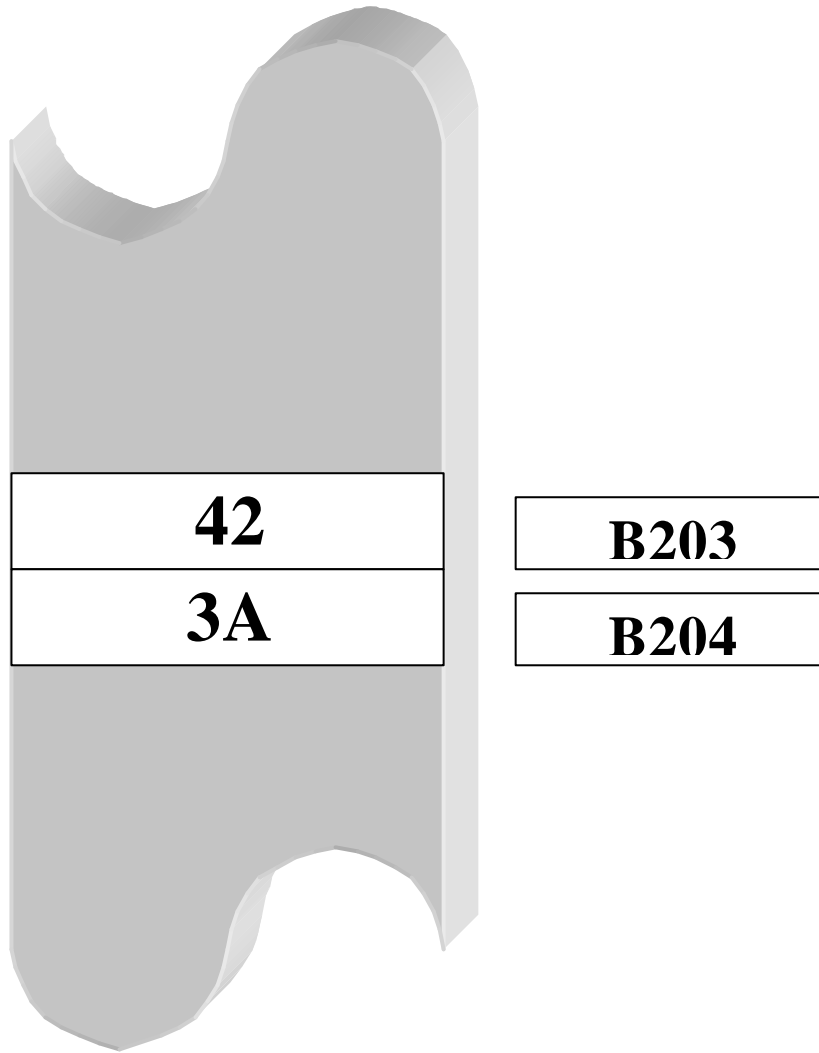
**CP (HL)**

**Prima**

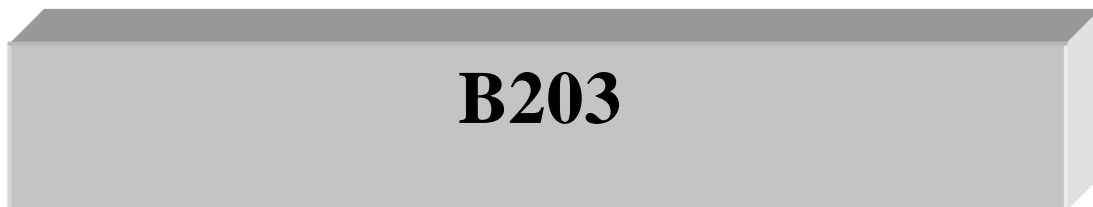
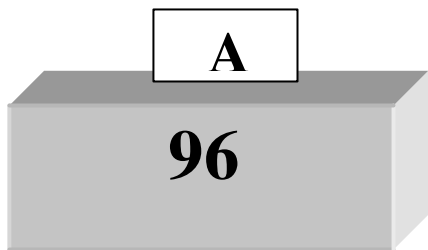


<b>S</b>	<b>Z</b>	<b>X</b>	<b>H</b>	<b>X</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>

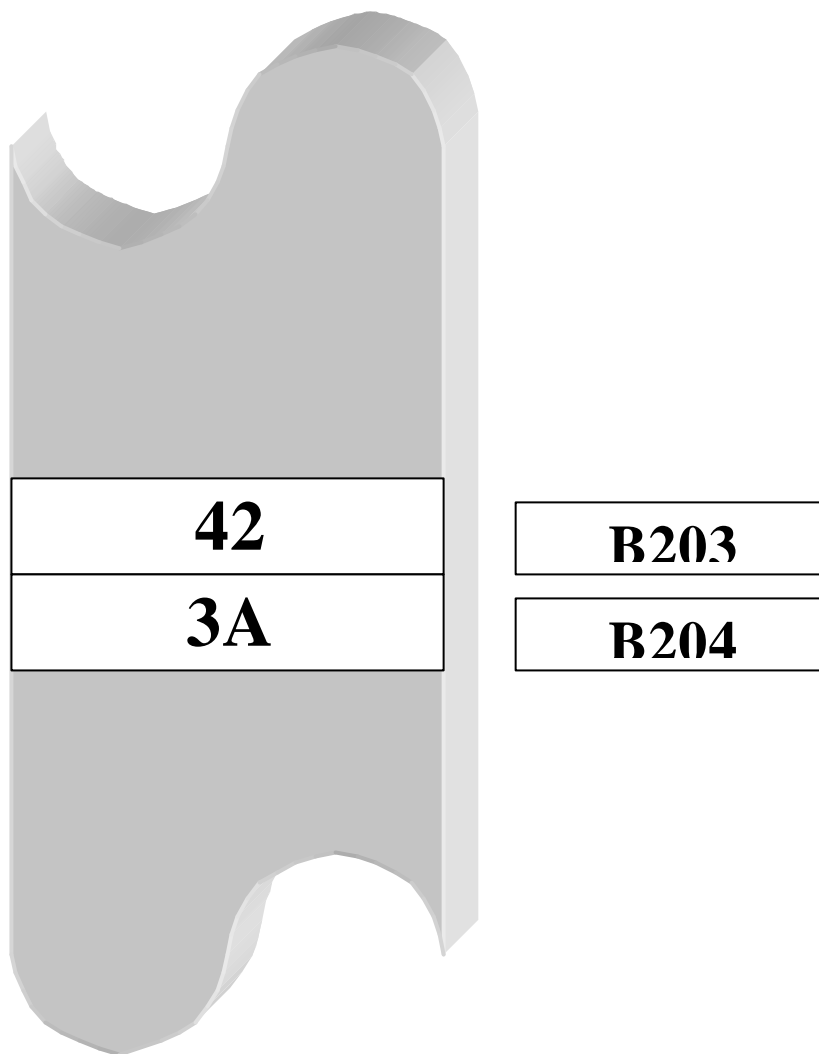
**Prima**



**Dopo**

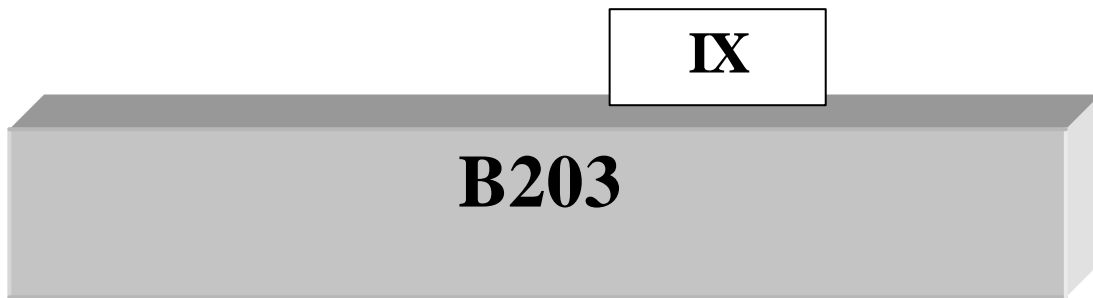
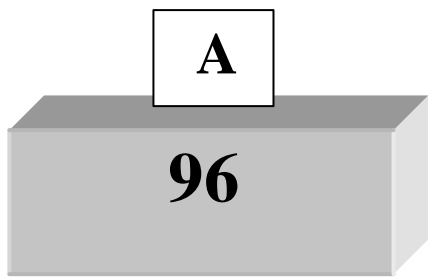


<b>S</b>	<b>Z</b>	<b>X</b>	<b>H</b>	<b>X</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>



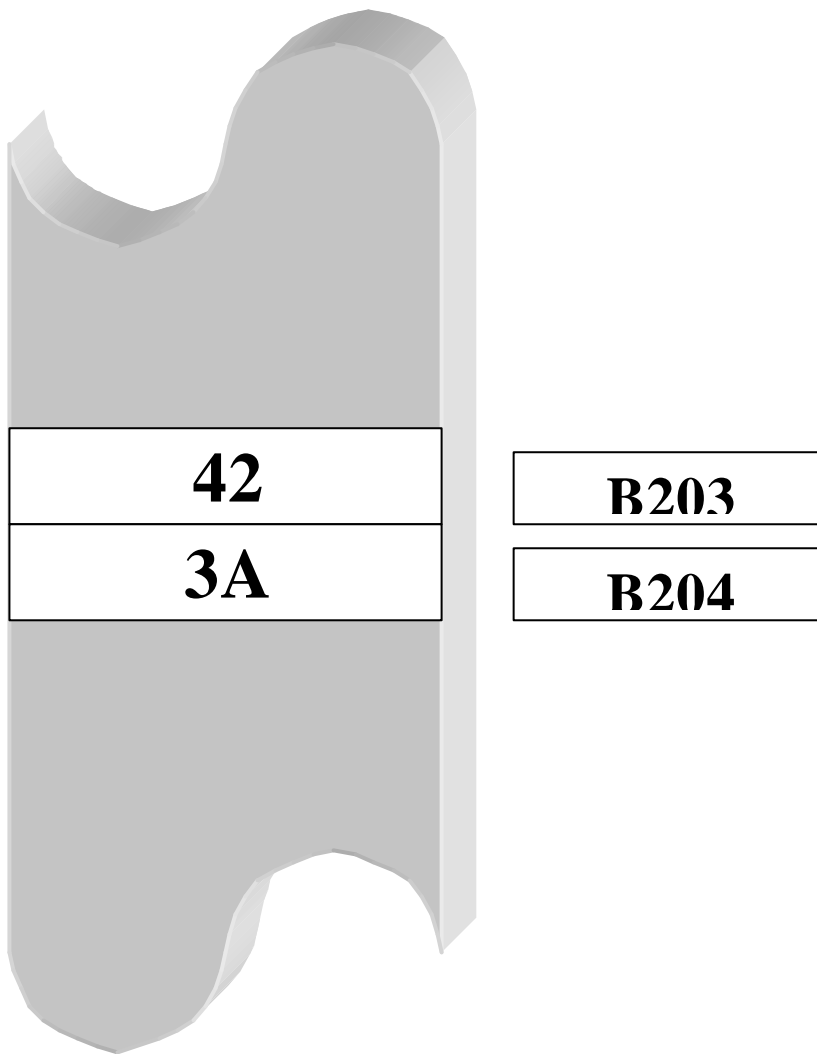
**Esempio**

**CP (IX+1)**

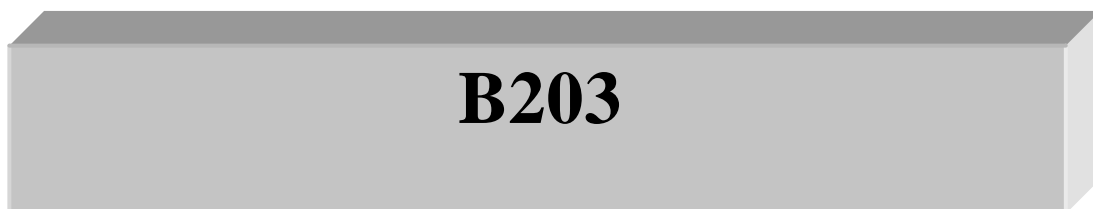
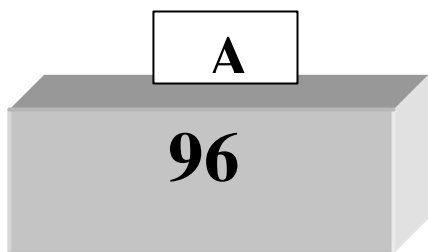


<b>S</b>	<b>Z</b>	<b>X</b>	<b>H</b>	<b>X</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>

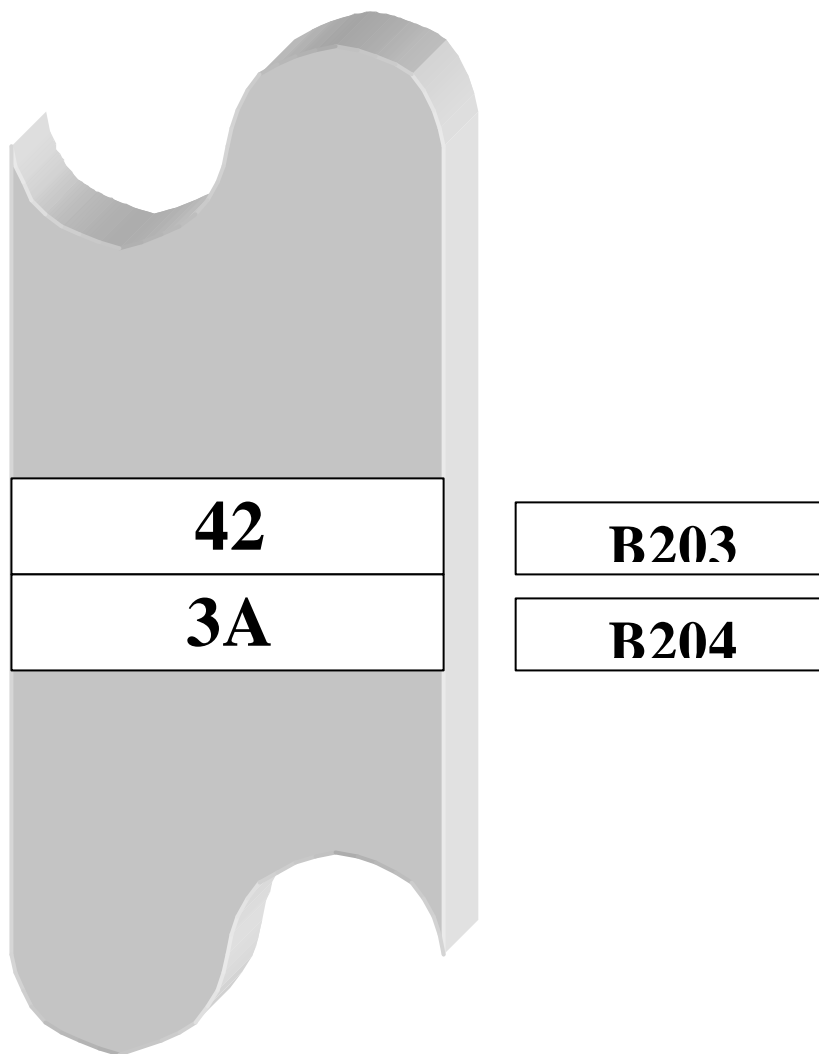
**Prima**



**Dopo**



<b>S</b>	<b>Z</b>	<b>X</b>	<b>H</b>	<b>X</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>



Per capire l'esempio ricordiamo che

$$96|_{16} = 10010110|_2$$

---

quindi rappresenta un numero negativo essendo l'MSB pari ad 1 mentre

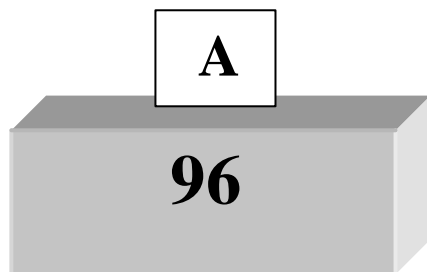
$$3A |_{16} = 00111010 |_2$$

quindi il risultato sarà negativo (S=1) e vi sarà un prestito dal nibble superiore al nibble inferiore (H=1)

**Esempio**

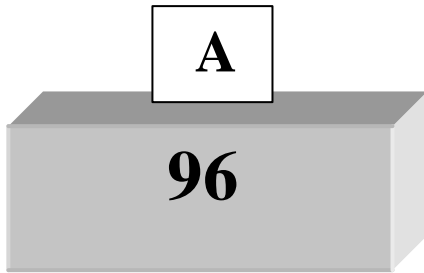
**CP 3A**

**Prima**



S	Z	X	H	X	P/V	N	C
0	0	0	1	0	1	1	0

Dopo



<b>S</b>	<b>Z</b>	<b>X</b>	<b>H</b>	<b>X</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>

**CPD**

Questa istruzione è molto utile quando si vuole ricercare un dato in un blocco di memoria. In A si pone il dato da cercare, l'accumulatore viene poi confrontato con le locazioni di memoria, il registro HL funge da puntatore poiché contiene di volta in volta l'indirizzo della locazione il cui contenuto si vuole confrontare con l'accumulatore, occorre poi un contatore in modo da sapere quando si è completato il confronto con tutte le locazioni di memoria appartenenti al blocco, il contatore è il registro BC. Con questa istruzione il confronto avviene partendo dalle locazioni di indirizzo più alto a decrescere.

***Quando viene eseguita quest'istruzione l'accumulatore viene confrontato con il contenuto della locazione il cui indirizzo è contenuto nel registro HL. Poi vengono decrementati il puntatore HL ed il contatore BC.***

**Modifiche al registro dei flag**

S	Z	H	P/V	N	C
●	<b>X</b>	●	<b>X</b>	<b>1</b>	●

Il flag Z ad uno indica che si è trovata una locazione il cui contenuto è uguale a quello dell'accumulatore. Il flag PV posto a zero indica che il contatore BC si è azzerato ed è quindi terminato il controllo del blocco di dati.

---

**Esempio Prima**

**BC**

**3154**

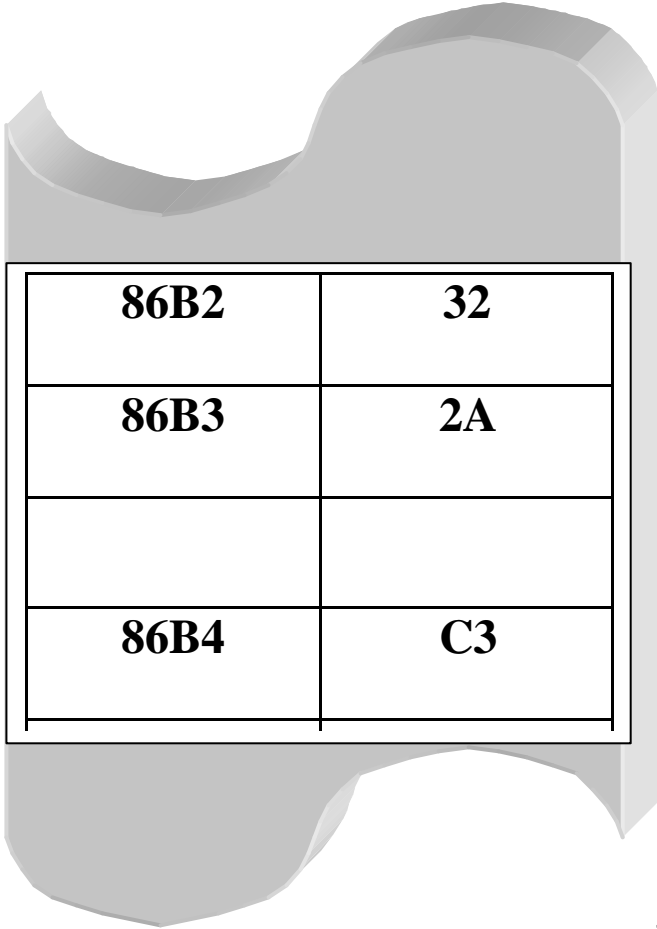
**A**

**2A**

**HL**

**86B5**

<b>S</b>	<b>Z</b>	<b>X</b>	<b>H</b>	<b>X</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>



**Dopo**

---

**BC**

**3153**

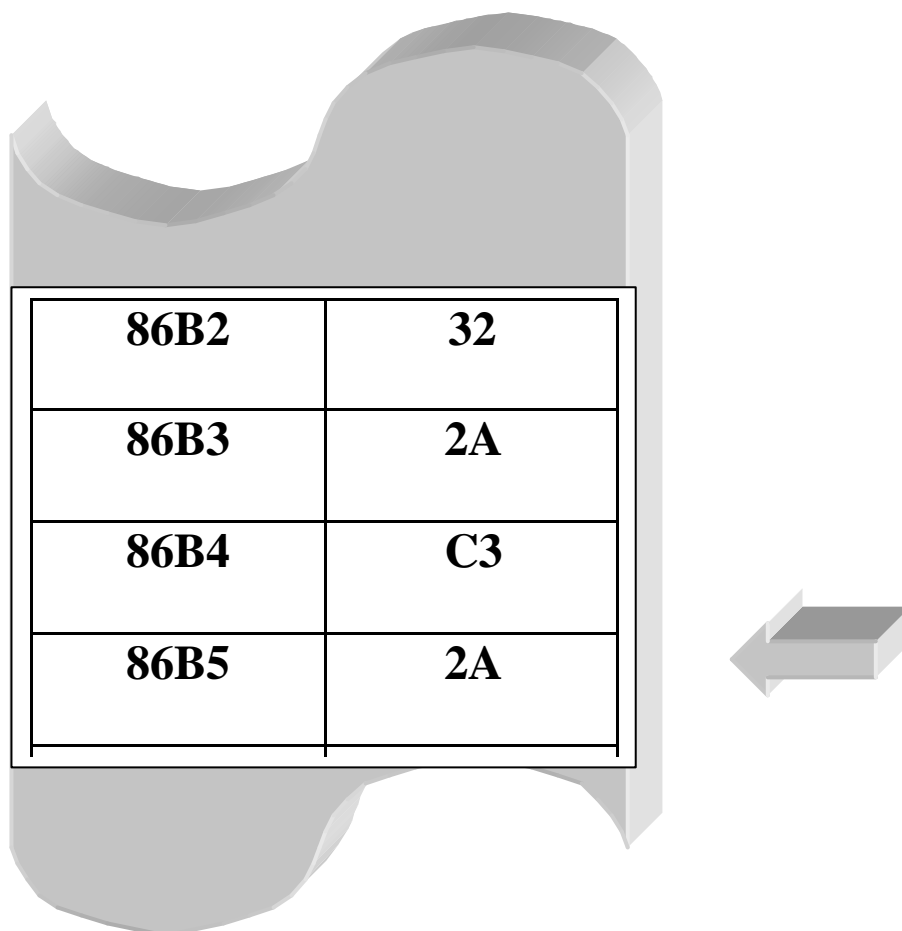
**A**

**2A**

**HL**

<b>S</b>	<b>Z</b>	<b>X</b>	<b>H</b>	<b>X</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>

Z va a uno poiché il dato contenuto nell'accumulatore coincide con il contenuto della locazione di memoria puntata da HL, PV è 1 poiché il registro BC non si è azzerato.



The diagram shows a stylized representation of a Z80 memory stack. A table is overlaid on the stack, showing the following data:

<b>86B2</b>	<b>32</b>
<b>86B3</b>	<b>2A</b>
<b>86B4</b>	<b>C3</b>
<b>86B5</b>	<b>2A</b>

To the right of the table, there is a 3D arrow pointing to the left, indicating a return or jump operation.

## **CPDR**

L'istruzione precedente necessita di essere inserita all'interno di un ciclo per poter essere eseguita su tutto il blocco di memoria in cui si vuole ricercare il dato. Alternativamente si può utilizzare l'istruzione CPDR, questa istruzione esegue tutte le azioni di CPD, ma a differenza di essa, le ripete fino a che si azzerava il contatore oppure appena trova una locazione contenente lo stesso dato dell'accumulatore.

**Modifiche al registro dei flag**

S	Z	H	P/V	N	C
●	<b>X</b>	●	<b>X</b>	<b>1</b>	●

Il flag Z è posto ad 1 se  $A=(HL)$

IL flag PV è posto a 0 se  $BC=0$  dopo l'esecuzione, altrimenti è posto ad 1

**CPI**

Questa istruzione è molto utile quando si vuole ricercare un dato in un blocco di memoria. In A si pone il dato da cercare, l'accumulatore viene poi confrontato con le locazioni di memoria, il registro HL funge da puntatore poiché contiene di volta in volta l'indirizzo della locazione il cui contenuto si vuole confrontare con l'accumulatore, occorre poi un contatore in modo da sapere quando si è completato il confronto con tutte le locazioni di memoria appartenenti al blocco, il contatore è il registro BC. Con questa istruzione il confronto avviene partendo dalle locazioni di indirizzo più basso a crescere.

***Quando viene eseguita quest'istruzione l'accumulatore viene confrontato con il contenuto della locazione il cui indirizzo è contenuto nel registro HL. Poi viene incrementato il puntatore HL e decrementato il contatore BC.***

**Modifiche al registro dei flag**

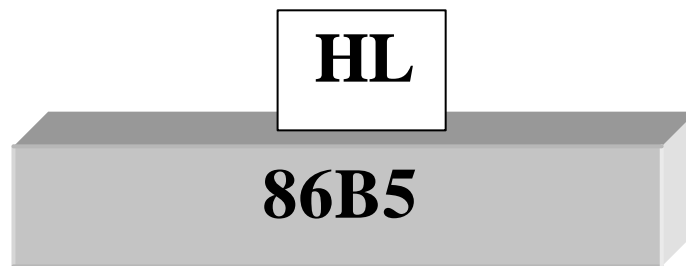
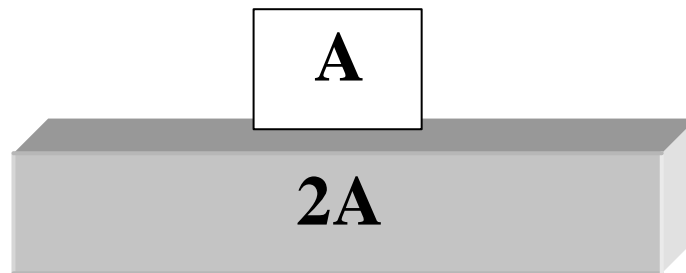
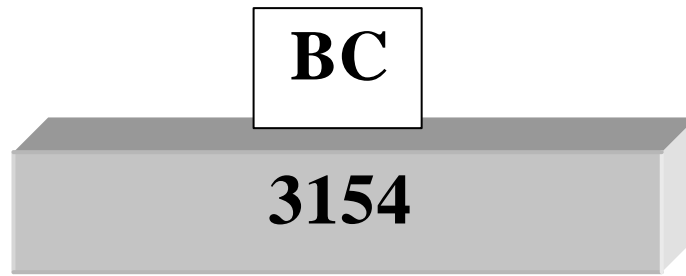
---

S	Z		H		P/V	N	C
●	<b>X</b>		●		<b>X</b>	<b>1</b>	●

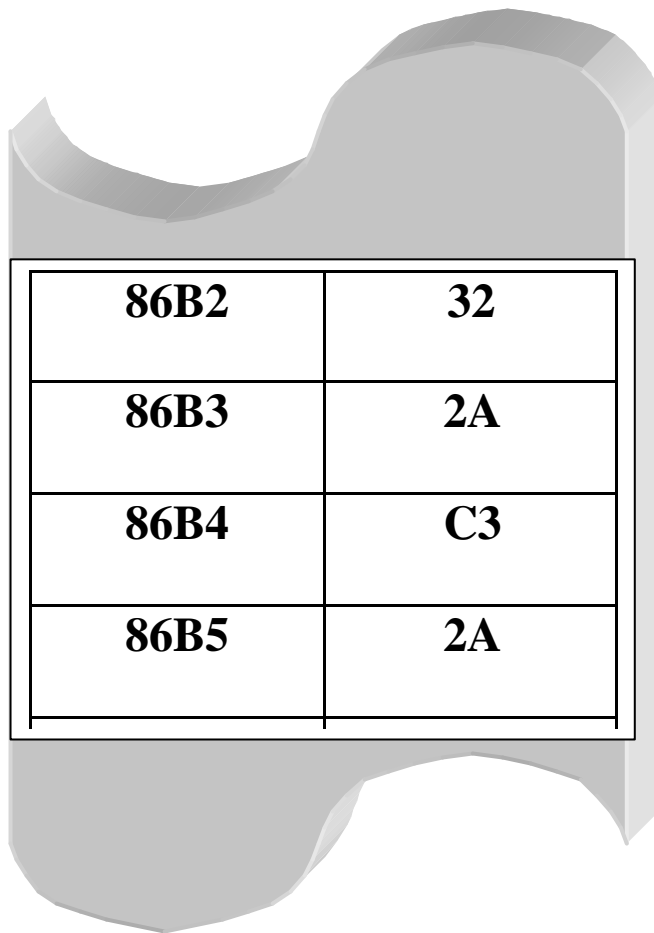
Il flag Z ad uno indica che si è trovata una locazione il cui contenuto è uguale a quello dell'accumulatore. Il flag PV posto a zero indica che il contatore BC si è azzerato ed è quindi terminato il controllo del blocco di dati.

---

**Esempio Prima**



<b>S</b>	<b>Z</b>	<b>X</b>	<b>H</b>	<b>X</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>



<b>86B2</b>	<b>32</b>
<b>86B3</b>	<b>2A</b>
<b>86B4</b>	<b>C3</b>
<b>86B5</b>	<b>2A</b>

**BC**

**3153**

**A**

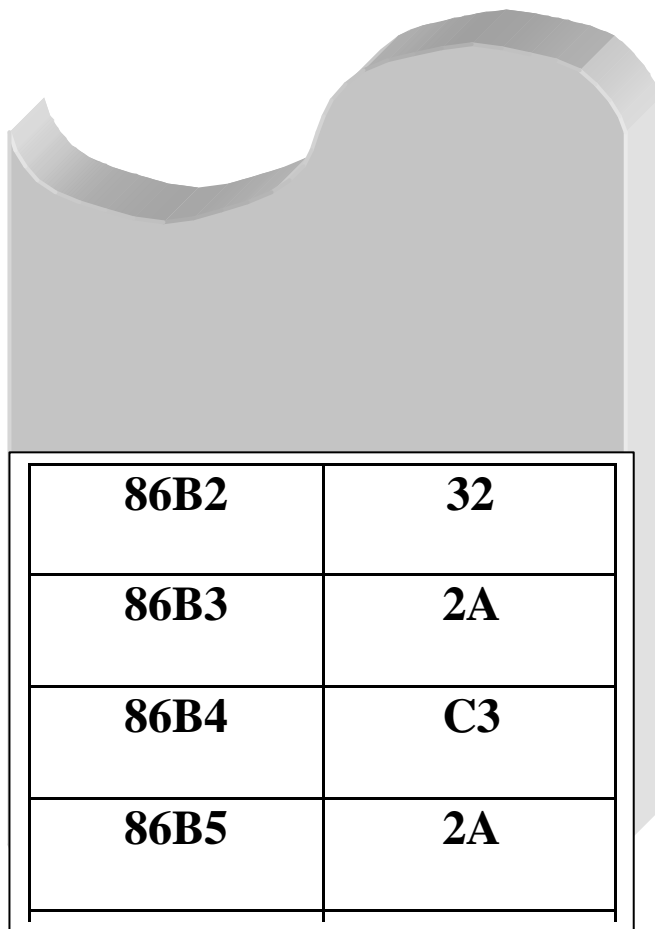
**2A**

**HL**

**86B6**

<b>S</b>	<b>Z</b>	<b>X</b>	<b>H</b>	<b>X</b>	<b>P/V</b>	<b>N</b>	<b>C</b>
<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>

**Dopo**



<b>86B2</b>	<b>32</b>
<b>86B3</b>	<b>2A</b>
<b>86B4</b>	<b>C3</b>
<b>86B5</b>	<b>2A</b>

Z va a uno poiché il dato contenuto nell'accumulatore coincide con il contenuto della locazione di memoria puntata da HL, PV è 1 poiché il registro BC non si è azzerato.

### CPIR

L'istruzione precedente necessita di essere inserita all'interno di un ciclo per poter essere eseguita su tutto il blocco di memoria in cui si vuole ricercare il dato. Alternativamente si può utilizzare l'istruzione CPIR, questa istruzione esegue tutte le azioni di CPD, ma a differenza di essa, le ripete fino a che si azzeri il contatore oppure appena trova una locazione contenente lo stesso dato dell'accumulatore.

### Modifiche al registro dei flag

S	Z		H		P/V	N	C
●	<b>X</b>		●		<b>X</b>	<b>1</b>	●

Il flag Z è posto ad 1 se A=(HL)

IL flag PV è posto a 0 se BC=0 dopo l'esecuzione, altrimenti è posto ad 1

*Istruzioni*

*Aritmetiche*

*E Logiche*

---

Quest'istruzione esegue la somma fra il contenuto dell'accumulatore, il secondo operando  $s$  ed il flag di carry

<b>ADC A, r</b>	Indirizzamento a registro	Si somma all'accumulatore il contenuto del registro ad otto bit $r$ (A, B, C, D, E, H, L)
<b>ADC A, n</b>	Indirizzamento immediato	Si somma all'accumulatore il contenuto del dato ad otto bit immediatamente espresso nell'istruzione
<b>ADC A, (HL)</b>	Indirizzamento indiretto	Si somma all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro HL
<b>ADC A, (IX+d)</b>	Indirizzamento indicizzato	Si somma all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IX sommato allo spiazzamento $d$ espresso nell'istruzione
<b>ADC A, (IY+d)</b>	Indirizzamento indicizzato	Si somma all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IY sommato allo spiazzamento $d$ espresso nell'istruzione

### Modifiche al registro dei flag



$$\begin{array}{r}
 + \\
 \begin{array}{|c|c|c|c|c|c|c|c|}
 \hline
 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
 \hline
 \end{array} \\
 + \\
 \begin{array}{|c|}
 \hline
 1 \\
 \hline
 \end{array} \\
 = \\
 \begin{array}{|c|c|c|c|c|c|c|c|}
 \hline
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
 \hline
 \end{array}
 \end{array}$$

Abbiamo il riporto fra il primo ed il secondo nibble, per cui H=1, il risultato è positivo per cui S=0, il risultato è non nullo per cui Z=0, non vi è un riporto sull'ottavo bit per cui C=0, il risultato rientra nel range -128/+127 per cui PV=0.

**Dopo**

A 

21
----

F 

11
----

### ADC HL, ss

Quest'istruzione esegue la somma fra il contenuto del registro a 16 bit HL, il secondo operando ss ed il flag di carry. Il secondo operando può essere soltanto un registro a 16 bit: BC, DE, HL, SP.

### Modifiche al registro dei flag

S	Z		H		P/V	N	C
●	●		●		●	0	●

- Il flag S verrà settato a seconda del segno del risultato
- Il flag Z verrà settato se il risultato è nullo
- Il flag H verrà settato se vi sarà un riporto fra l'undicesimo ed il dodicesimo bit
- Il flag PV verrà settato se vi sarà un errore di overflow
- Il flag N è a zero trattandosi di una somma
- Il flag C verrà settato se vi è un riporto sul sedicesimo bit

**Esempio:**

**ADC HL, DE**

**Prima**

41

F

3291

DE

0F18

HL

**Dopo**

41

F

3291

DE

41AA

HL

**ADD A, s**

Quest'istruzione esegue la somma fra il contenuto dell'accumulatore e il secondo operando s.

**ADD A, r** Indirizzamento a regi- Si somma all'accumulatore il

---

	stro	contenuto del registro ad otto bit r (A, B, C, D, E, H, L)
<b>ADD A, n</b>	Indirizzamento immediato	Si somma all'accumulatore il contenuto del dato ad otto bit immediatamente espresso nell'istruzione
<b>ADD A, (HL)</b>	Indirizzamento indiretto	Si somma all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro HL
<b>ADD A, (IX+d)</b>	Indirizzamento indicizzato	Si somma all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IX sommato allo spiazzamento d espresso nell'istruzione
<b>ADD A, (IY+d)</b>	Indirizzamento indicizzato	Si somma all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IY sommato allo spiazzamento d espresso nell'istruzione



---

E' stata effettuata la somma fra il contenuto dell'accumulatore pari a  $02_{\text{h}}$  ed il contenuto della locazione d'indirizzo  $9620_{\text{h}}$  (contenuto in HL) pari a  $B1_{\text{h}}$

$$\mathbf{02} \mid_{16} = \mathbf{00000010} \mid_{10} +$$

$$\mathbf{B1} \mid_{16} = \mathbf{10110001} \mid_{10} =$$

---

$$\mathbf{B3} \mid_{16} = \mathbf{10110011} \mid_{10}$$

Il risultato è negativo ( $S=1$ ), diverso da zero ( $Z=0$ ), non vi è stato un riporto fra il nibble inferiore ed il nibble superiore ( $H=0$ ), senza overflow ( $PV=0$ ), senza riporto ( $C=0$ ). Essendo un'addizione  $N=0$ .

80
----

F

**ADD HL, ss**

Quest'istruzione esegue la somma fra il contenuto del registro a 16 bit HL ed il secondo operando ss. Il secondo operando può essere soltanto un registro a 16 bit: BC, DE, HL, SP.

**Modifiche al registro dei flag**

S	Z		H		P/V	N	C
●	●		●		●	0	●

- Il flag S verrà settato a seconda del segno del risultato
- Il flag Z verrà settato se il risultato è nullo
- Il flag H verrà settato se vi sarà un riporto fra l'undicesimo ed il dodicesimo bit
- Il flag PV verrà settato se vi sarà un errore di overflow
- Il flag N è a zero trattandosi di una somma
- Il flag C verrà settato se vi è un riporto sul sedicesimo bit

**Esempio:****ADD HL, DE****Prima**

3291	DE	0F18	HL
------	----	------	----

**Dopo**

3291	DE	41A9	HL
------	----	------	----

**ADD IX, rr**

Quest'istruzione esegue la somma fra il contenuto del registro a 16 bit IX ed il secondo operando ss. Il secondo operando può essere soltanto un registro a 16 bit: BC, DE, HL, SP.

**Modifiche al registro dei flag**

S	Z	H	P/V	N	C
		●		0	●

- Il flag S non viene modificato
- Il flag Z non viene modificato
- Il flag H verrà settato se vi sarà un riporto fra l'undicesimo ed il dodicesimo bit
- Il flag PV non viene modificato
- Il flag N è a zero trattandosi di una somma
- Il flag C verrà settato se vi è un riporto sul sedicesimo bit

**Esempio:** **ADD IX, SP**

**Prima**

0000	IX	3021	SP
------	----	------	----

**Dopo**

3021	IX	3021	SP
------	----	------	----

**ADD IY, rr**

Quest'istruzione esegue la somma fra il contenuto del registro a 16 bit IY ed il secondo operando ss. Il secondo operando può essere soltanto un registro a 16 bit: BC, DE, HL, SP.

**Modifiche al registro dei flag**

S	Z	H	P/V	N	C
		●		0	●

- Il flag S non viene modificato
- Il flag Z non viene modificato
- Il flag H verrà settato se vi sarà un riporto fra l'undicesimo ed il dodicesimo bit
- Il flag PV non viene modificato
- Il flag N è a zero trattandosi di una somma
- Il flag C verrà settato se vi è un riporto sul sedicesimo bit

**Esempio:                      ADD IY, SP****Prima**

0000	IY	3021	SP
------	----	------	----

**Dopo**

3021	IY	3021	SP
------	----	------	----

**INC s**

Quest'istruzione esegue l'incremento di un'unità del contenuto dell'operando s.

<b>INC r</b>	Indirizzamento a registro	Si incrementa il contenuto del registro ad otto bit r (A, B, C, D, E, H, L)
<b>INC (HL)</b>	Indirizzamento indiretto	Si incrementa il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro HL
<b>INC (IX+d)</b>	Indirizzamento indicizzato	Si incrementa il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IX sommato allo spazamento d espresso nell'istruzione
<b>INC (IY+d)</b>	Indirizzamento indicizzato	Si incrementa il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IY sommato allo spazamento d espresso nell'istruzione

**Modifiche al registro dei flag**

<b>S</b>	<b>Z</b>		<b>H</b>		<b>P/V</b>	<b>N</b>	<b>C</b>
●	●		●		●	0	

- Il flag S verrà settato a seconda del segno del risultato
- Il flag Z verrà settato se il risultato è nullo
- Il flag H verrà settato se vi sarà un riporto fra il quarto e il quinto bit
- Il flag PV verrà settato se vi sarà un errore di overflow
- Il flag N è a zero trattandosi di una somma
- Il flag C non viene modificato

**INC rr**

Quest'istruzione esegue l'incremento di un'unità del contenuto del registro a 16 bit rr. Esso può essere BC, DE, HL, SP, IX, IY

***Non ha effetto sul registro dei flag***

**SBC A, s**

Quest'istruzione esegue la sottrazione fra il contenuto dell'accumulatore, il secondo operando s ed il flag di carry

<b>SBC A, r</b>	Indirizzamento a registro	Si sottrae all'accumulatore il contenuto del registro ad otto bit r (A, B, C, D, E, H, L)
<b>SBC A, n</b>	Indirizzamento immediato	Si sottrae all'accumulatore il contenuto del dato ad otto bit immediatamente espresso nell'istruzione
<b>SBC A, (HL)</b>	Indirizzamento indiretto	Si sottrae all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro HL
<b>SBC A, (IX+d)</b>	Indirizzamento indicizzato	Si sottrae all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IX sommato allo spiazzamento d espresso nell'istruzione
<b>SBC A,(IY+d)</b>	Indirizzamento indicizzato	Si sottrae all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IY sommato allo spiazzamento d espresso nell'istruzione

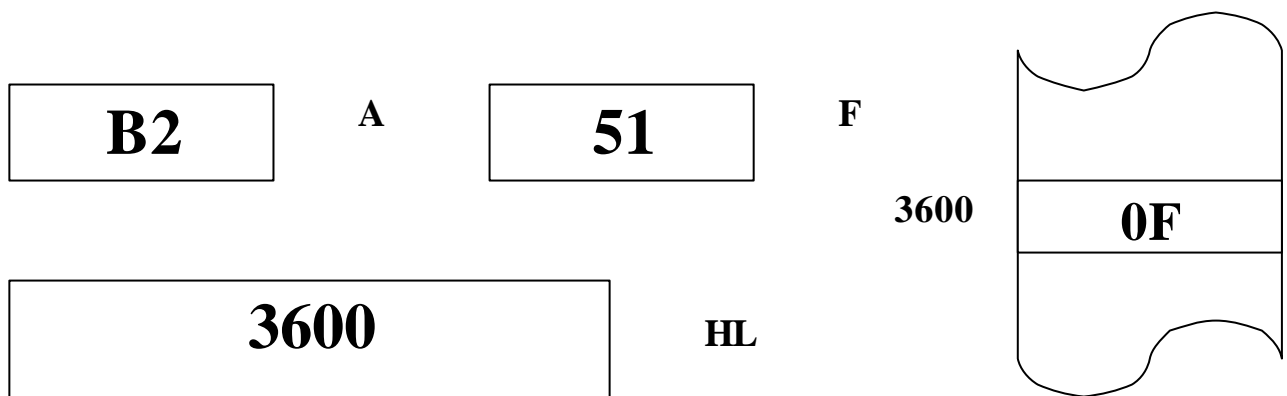
**Modifiche al registro dei flag**

S	Z		H		P/V	N	C
●	●		●		●	<b>1</b>	●

- Il flag S verrà settato a seconda del segno del risultato
- Il flag Z verrà settato se il risultato è nullo
- Il flag H verrà settato se vi sarà un prestito fra il quinto e il quarto bit
- Il flag PV verrà settato se vi sarà un errore di overflow
- Il flag N è a uno trattandosi di una sottrazione
- Il flag C verrà settato se vi è un prestito sull'ottavo bit

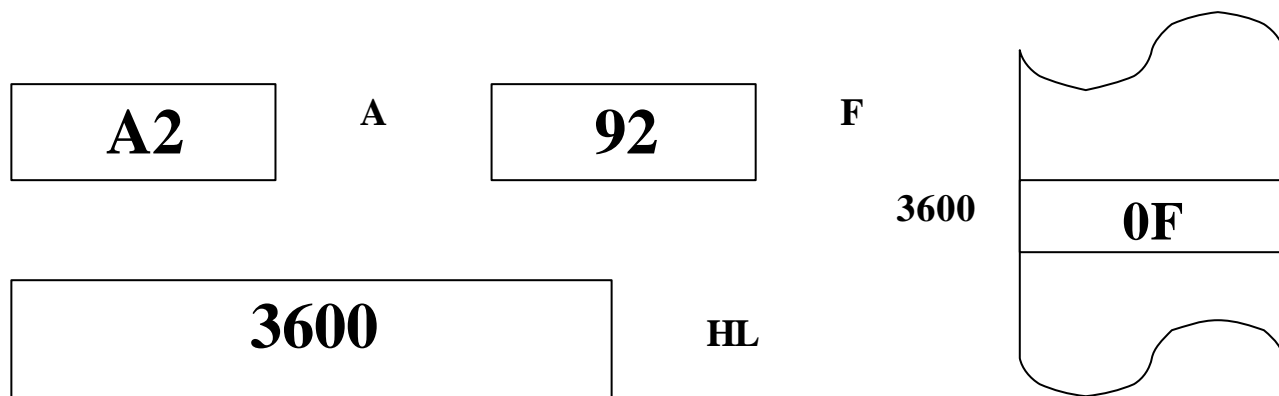
**Esempio: SBC A, (HL)**

**Prima**



---

**Dopo**



Infatti viene effettuata la sottrazione

$$\begin{array}{r} \mathbf{B2} - \\ \mathbf{0F} - \\ \mathbf{1} = \\ \mathbf{A2} \end{array}$$

Questa sottrazione comporta un prestito dal nibble superiore al nibble inferiore per cui sarà  $H=1$ , inoltre il risultato sarà negativo (infatti  $A2 |_{16} = 10100010$  quindi l'MSB è pari ad uno per cui è la rappresentazione di un numero negativo)

**SUB A, s**

Quest'istruzione esegue la sottrazione fra il contenuto dell'accumulatore ed il secondo operando s

<b>SUB A, r</b>	Indirizzamento a registro	Si sottrae all'accumulatore il contenuto del registro ad otto bit r (A, B, C, D, E, H, L)
<b>SUB A, n</b>	Indirizzamento immediato	Si sottrae all'accumulatore il contenuto del dato ad otto bit immediatamente espresso nell'istruzione
<b>SUB A, (HL)</b>	Indirizzamento indiretto	Si sottrae all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro HL
<b>SUB A, (IX+d)</b>	Indirizzamento indicizzato	Si sottrae all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IX sommato allo spiazzamento d espresso nell'istruzione
<b>SUB A,(IY+d)</b>	Indirizzamento indicizzato	Si sottrae all'accumulatore il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IY sommato allo spiazzamento d espresso nell'istruzione

---

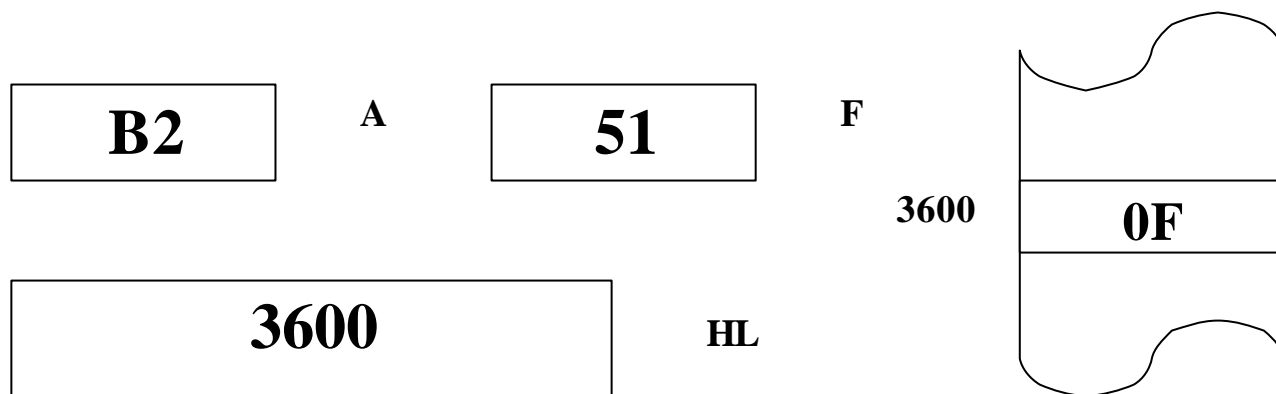
**Modifiche al registro dei flag**

S	Z		H		P/V	N	C
●	●		●		●	<b>1</b>	●

- Il flag S verrà settato a seconda del segno del risultato
- Il flag Z verrà settato se il risultato è nullo
- Il flag H verrà settato se vi sarà un prestito fra il quinto e il quarto bit
- Il flag PV verrà settato se vi sarà un errore di overflow
- Il flag N è a uno trattandosi di una sottrazione
- Il flag C verrà settato se vi è un prestito sull'ottavo bit

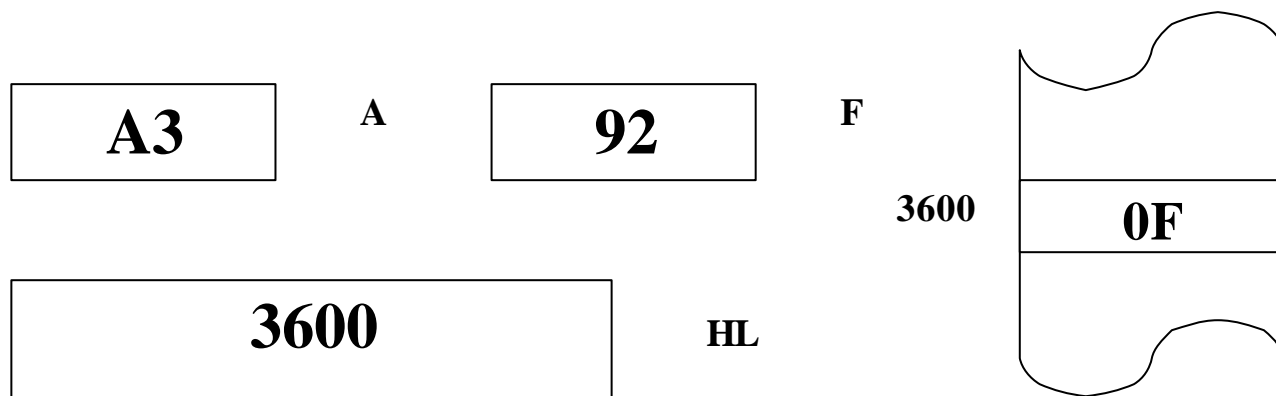
**Esempio:**                      **SUB A, (HL)**

**Prima**



---

**Dopo**



Infatti viene effettuata la sottrazione

**B2-**

**0F=**

**A3**

Questa sottrazione comporta un prestito dal nibble superiore al nibble inferiore per cui sarà  $H=1$ , inoltre il risultato sarà negativo (infatti  $A3|_{16}=10100011$  quindi l'MSB è pari ad uno per cui è la rappresentazione di un numero negativo)

---

**SBC HL, ss**

Questa istruzione esegue la sottrazione fra il contenuto del registro HL, il contenuto del registro a 16 bit ss ed il flag di Carry. Il risultato viene memorizzato in HL

$$HL \leftarrow HL - ss - FC$$

**Modifiche al registro dei flag**

S	Z		H		P/V	N	C
●	●		●		●	<b>1</b>	●

- Il flag S verrà settato a seconda del segno del risultato
- Il flag Z verrà settato se il risultato è nullo
- Il flag H verrà settato se vi sarà un prestito fra il terdicesimo e il dodicesimo bit
- Il flag PV verrà settato se vi sarà un errore di overflow
- Il flag N è a uno trattandosi di una sottrazione
- Il flag C verrà settato se vi è un prestito sul sedicesimo bit

---

**Esempio:** **SBC HL, DE**

**Prima**

66
----

**F**

06B9
------

**DE**

3142
------

**HL**

**Dopo**

12
----

**F**

06B9
------

**DE**

2A79
------

**HL**

Poiché inizialmente nel registro dei flag vi è un numero pari (66) vuol dire che l'LSB corrispondente al flag di carry è pari a zero. Il risultato è un numero positivo (S=0, Z=0), vi è un prestito dal quarto al terzo nibble (H=1), N=1 poiché è una sottrazione.

**DEC s**

Quest'istruzione esegue il decremento di un'unità del contenuto dell'operando s.

<b>DEC r</b>	Indirizzamento a registro	Si decrementa il contenuto del registro ad otto bit r (A, B, C, D, E, H, L)
<b>DEC (HL)</b>	Indirizzamento indiretto	Si decrementa il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro HL
<b>DEC (IX+d)</b>	Indirizzamento indicizzato	Si decrementa il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IX sommato allo spazamento d espresso nell'istruzione
<b>DEC (IY+d)</b>	Indirizzamento indicizzato	Si decrementa il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IY sommato allo spazamento d espresso nell'istruzione

**Modifiche al registro dei flag**

<b>S</b>	<b>Z</b>		<b>H</b>		<b>P/V</b>	<b>N</b>	<b>C</b>
●	●		●		●	<b>1</b>	

- Il flag S verrà settato a seconda del segno del risultato
- Il flag Z verrà settato se il risultato è nullo
- Il flag H verrà settato se vi sarà un prestito fra il quarto e il quinto bit
- Il flag PV verrà settato se vi sarà un errore di overflow
- Il flag N è a 1 trattandosi di una sottrazione
- Il flag C non viene modificato

**DEC rr**

Quest'istruzione esegue il decremento di un'unità del contenuto del registro a 16 bit rr. Esso può essere BC, DE, HL, SP, IX, IY

***Non ha effetto sul registro dei flag***

**DAA**

E' l'istruzione di aggiustamento decimale dell'accumulatore. Il suo scopo è quello di trasformare il contenuto binario dell'accumulatore in un numero BCD.

Per comprendere quest'operazione ricordiamo che il codice BCD si differenzia da quello binario poiché un nibble di quattro bit , pur potendo costituire teoricamente 16 combinazioni diverse, ne sfrutta soltanto le prime 10 per rappresentare soltanto le cifre decimali da 0 a 9. A differenza del codice binario, quindi, non sono consentite le combinazioni corrispondenti alle cifre da A ad F.

**AND s**

Viene effettuato l'And logico fra accumulatore ed operando s ed il risultato viene immagazzinato nell'accumulatore. And logico significa che ogni bit dell'accumulatore viene sostituito dall'and fra esso ed il suo corrispondente nell'operando s

<b>AND A, r</b>	Indirizzamento a registro	L'operando s è il contenuto del registro ad otto bit r (A, B, C, D, E, H, L)
<b>AND A, n</b>	Indirizzamento immediato	L'operando s è il contenuto del dato ad otto bit immediatamente espresso nell'istruzione
<b>AND A, (HL)</b>	Indirizzamento indiretto	L'operando s è il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro HL
<b>AND A, (IX+d)</b>	Indirizzamento indicizzato	L'operando s è il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IX sommato allo spazamento d espresso nell'istruzione
<b>AND A, (IY+d)</b>	Indirizzamento indicizzato	L'operando s è il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IY sommato allo spazamento d espresso nell'istruzione



**OR s**

Viene effettuato l'or logico fra accumulatore ed operando s ed il risultato viene immagazzinato nell'accumulatore. Or logico significa che ogni bit dell'accumulatore viene sostituito dall'or fra esso ed il suo corrispondente nell'operando s

<b>OR A, r</b>	Indirizzamento a registro	L'operando s è il contenuto del registro ad otto bit r (A, B, C, D, E, H, L)
<b>OR A, n</b>	Indirizzamento immediato	L'operando s è il contenuto del dato ad otto bit immediatamente espresso nell'istruzione
<b>OR A, (HL)</b>	Indirizzamento indiretto	L'operando s è il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro HL
<b>OR A, (IX+d)</b>	Indirizzamento indicizzato	L'operando s è il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IX sommato allo spazamento d espresso nell'istruzione
<b>OR A, (IY+d)</b>	Indirizzamento indicizzato	L'operando s è il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IY sommato allo spazamento d espresso nell'istruzione



**XOR s**

Viene effettuato l'or esclusivo logico fra accumulatore ed operando s ed il risultato viene immagazzinato nell'accumulatore. XOr logico significa che ogni bit dell'accumulatore viene sostituito dall'or esclusivo fra esso ed il suo corrispondente nell'operando s

<b>XOR A, r</b>	Indirizzamento a registro	L'operando s è il contenuto del registro ad otto bit r (A, B, C, D, E, H, L)
<b>XOR A, n</b>	Indirizzamento immediato	L'operando s è il contenuto del dato ad otto bit immediatamente espresso nell'istruzione
<b>XOR A, (HL)</b>	Indirizzamento indiretto	L'operando s è il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro HL
<b>XOR A,(IX+d)</b>	Indirizzamento indicizzato	L'operando s è il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IX sommato allo spazamento d espresso nell'istruzione
<b>XOR A,(IY+d)</b>	Indirizzamento indicizzato	L'operando s è il contenuto della locazione di memoria il cui indirizzo è dato dal contenuto del registro IY sommato allo spazamento d espresso nell'istruzione

**Modifiche al registro dei flag**

S	Z		H		P/V	N	C
●	●		0		●	0	0

- Il flag S verrà settato a seconda del segno del risultato
- Il flag Z verrà settato se il risultato è nullo
- Il flag H verrà posto sempre a zero
- Il flag PV verrà settato ad uno se il numero di bit ad 1 nel risultato è pari altrimenti è a zero
- Il flag N è sempre a zero
- Il flag C è sempre a zero

**Esempio:** **XOR 4B**

**Prima**

<b>36</b>
-----------

A

**36** |<sub>16</sub>=**00110110** |<sub>2</sub>

**XOR**

**4B** |<sub>16</sub>=**01001011** |<sub>2</sub>

=

**02** |<sub>16</sub>=**01111101** |<sub>2</sub>

**Dopo**

<b>7D</b>
-----------

A

**CPL**

L'accumulatore viene complementato ad uno, cioè ogni suo singolo bit viene invertito.

**Modifiche al registro dei flag**

S	Z	H	P/V	N	C
		<b>1</b>		<b>1</b>	

**Esempio**

**Prima**

<b>36</b>
-----------

A

$$36 |_{16} = 00110110 |_2$$

**COMPLEMENTA**

$$C9 |_{16} = 11001001 |_2$$

<b>C9</b>
-----------

A

**NEG**

L'accumulatore viene complementato a due, poiché viene sottratto a zero

**Modifiche al registro dei flag**

S	Z		H		P/V	N	C
●	●		●		●	<b>1</b>	●

- Il flag S verrà settato a seconda del segno del risultato
- Il flag Z verrà settato se il risultato è nullo
- Il flag H verrà settato se vi sarà un prestito fra il quarto e il quinto bit
- Il flag PV verrà settato se A era 80h prima dell'esecuzione dell'istruzione. Infatti  $80|_{16} = -128$ , ma  $0 - (-128) = +128 > +127$  quindi si va in overflow
- Il flag N è a 1
- Il flag C sarà posto ad 1 se A era nullo prima dell'esecuzione dell'istruzione. Infatti proviamo ad effettuare l'esecuzione del complemento a due :  $00000000|_2 \rightarrow$  (complemento ad uno)  $11111111|_2 \rightarrow 00000000|_2$  con un riporto sull'ottavo bit

**Esempio****Prima**

<b>32</b>
-----------

A

**$32|_{16} = 00110010|_2 \rightarrow$  (COMPLEMENTA AD UNO CD)  $|_{16} = 11001101|_2 \rightarrow$   
(COMPLEMENTA A DUE)  $11001101|_2 + 1 = 11001110|_2 = CE|_{16}$**

**Dopo**

**C9**

**A**