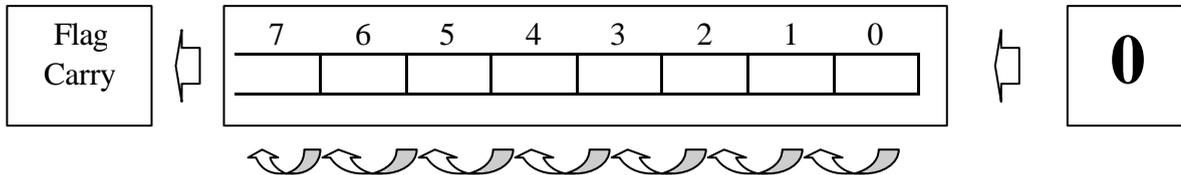


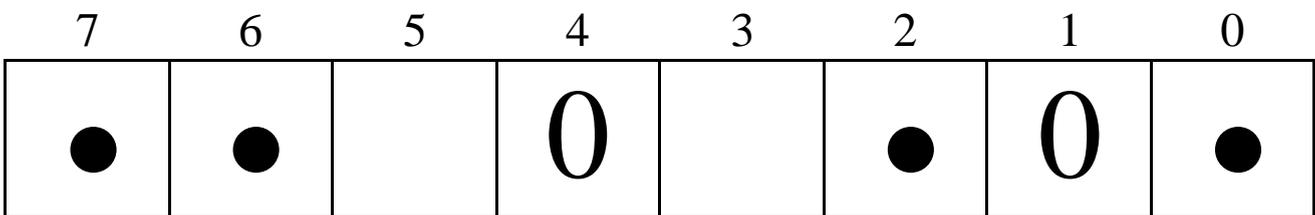
ISTRUZIONI DI SHIFT

SLA m

Quest'operazione effettua lo shift a sinistra del dato m: ogni bit viene spostato al posto del bit che si trova alla sua sinistra. Al posto del bit zero viene inserito uno zero. Il vecchio contenuto del bit sette viene posto nel flag di carry.



Modifica del registro dei flag

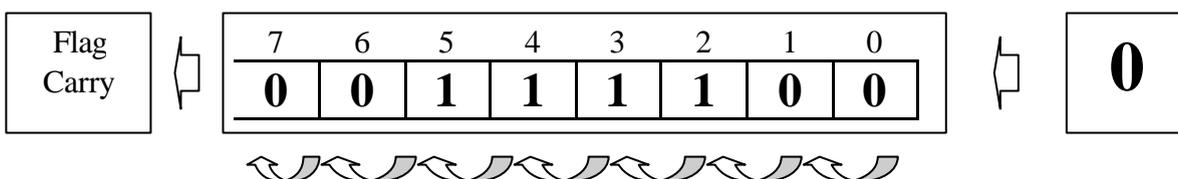


- Il flag S funziona come flag di segno, e rappresenta il segno del dato dopo lo shift
- Il flag Z funziona come flag di zero
- H viene posto sempre a zero
- Il flag PV funziona come flag di parità
- Il flag N è posto sempre a zero
- Il flag di carry conterrà il bit sette del dato

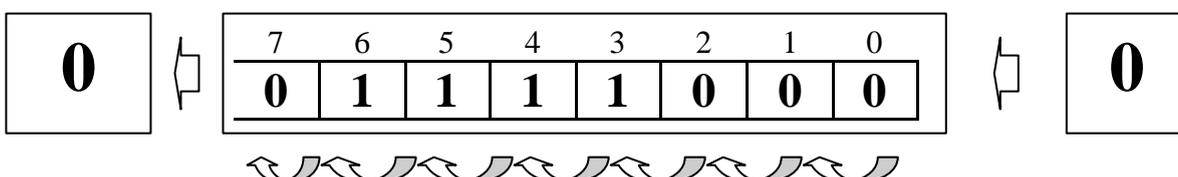
✓ L'operando m può essere un registro ad otto bit dello Z80

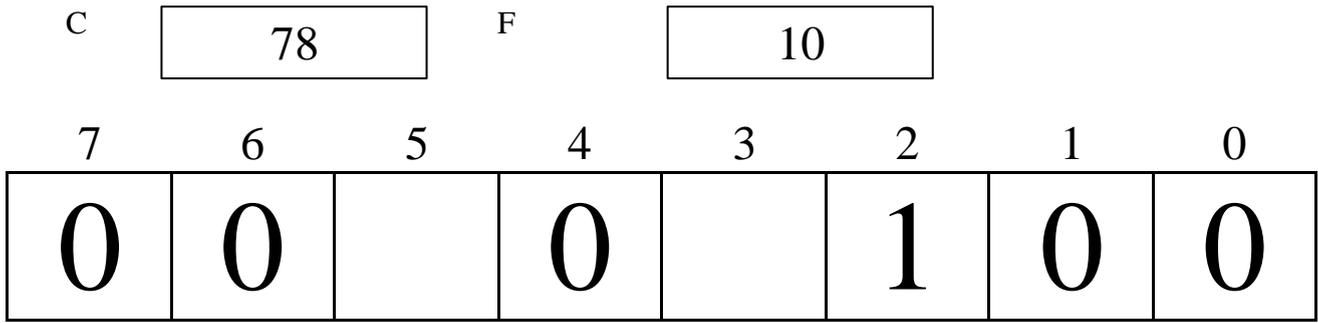
Es. SLA C

Prima



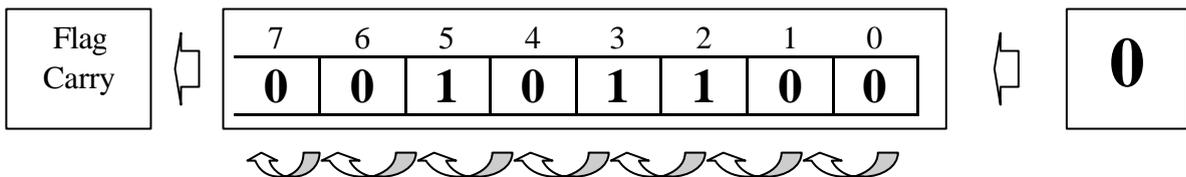
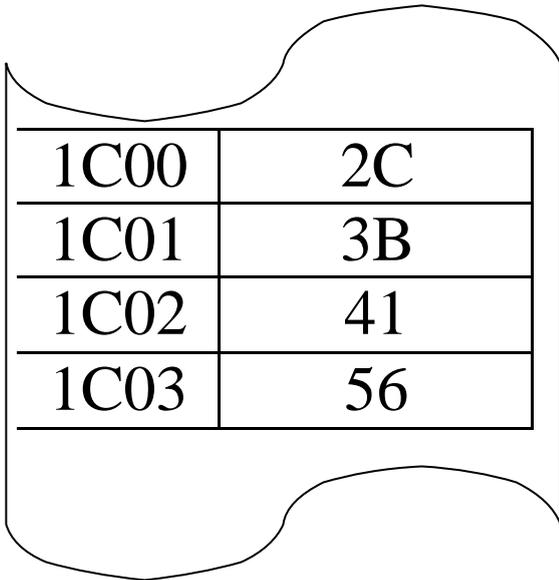
Dopo



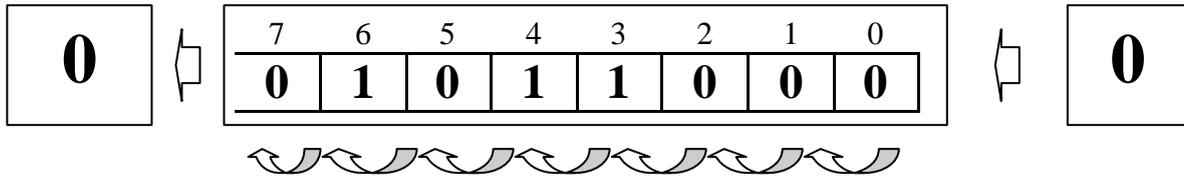


- Il risultato dello shift è stato un numero positivo (S=0)
 - Il risultato dello shift è stato un numero diverso da zero (Z=0)
 - Il risultato dello shift presenta un numero di bit ad uno pari a quattro (numero pari) per cui PV=1
 - Il bit sette del dato prima dello shift era zero per cui C = 0
 - ✓ L'operando m può essere un operando di memoria il cui indirizzo si trova nel registro HL (indirizzato in modo indiretto)
- Es. SLA (HL)

Prima

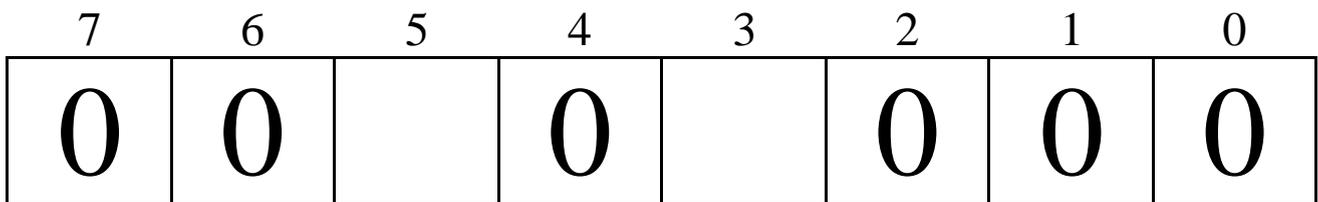


Dopo



HL 1C00 F 00

1C00	58
1C01	3B
1C02	41
1C03	56



- Il risultato dello shift è stato un numero positivo (S=0)
- Il risultato dello shift è stato un numero diverso da zero (Z=0)
- Il risultato dello shift presenta un numero di bit ad uno pari a tre (numero pari) per cui PV=0
- Il bit sette del dato prima dello shift era zero per cui C = 0

✓ L'operando m può essere il contenuto di una locazione di memoria il cui indirizzo si ottiene sommando il contenuto di un registro indice (IX o IY) ad uno spiazzamento (indirizzamento indicizzato)

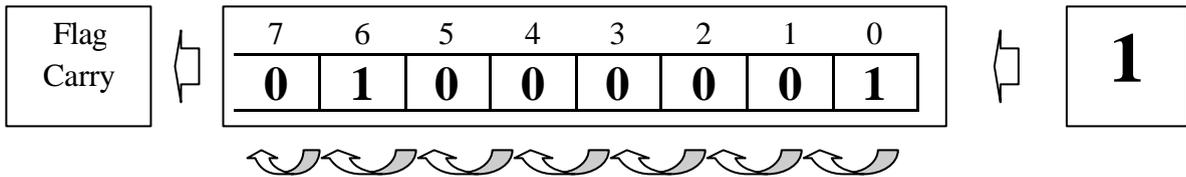
Es. SLA (IX+2)

Prima

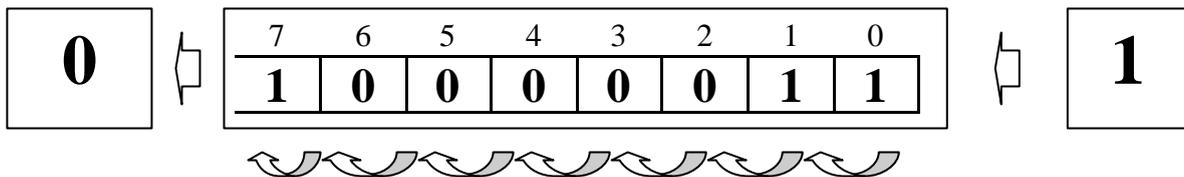
IX 1C00

F 11

1C00	2C
1C01	3B
1C02	41
1C03	56



Dopo



HL 1C00

F 10

1C00	2C
1C01	3B
1C02	83
1C03	56

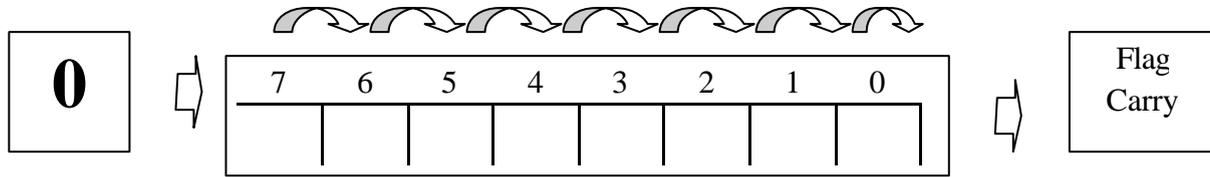
7	6	5	4	3	2	1	0
1	0		0		0	0	0

- Il risultato dello shift è stato un numero negativo ($S=1$)
- Il risultato dello shift è stato un numero diverso da zero ($Z=0$)
- Il risultato dello shift presenta un numero di bit ad uno pari a tre (numero dispari) per cui $PV=0$
- Il bit sette del dato prima dello shift era zero per cui $C = 0$

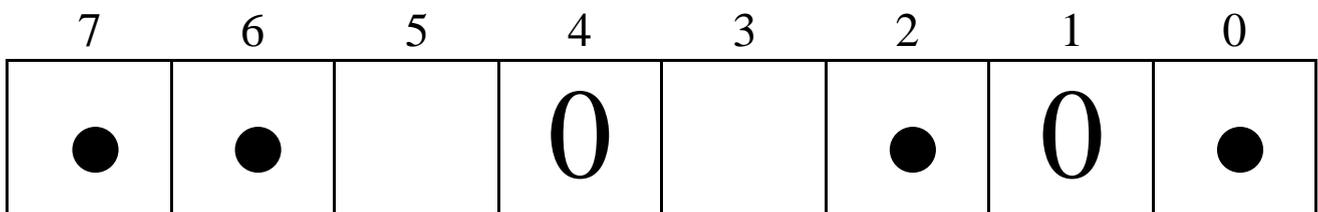
Quest'istruzione viene chiamata shift a sinistra aritmetico poiché uno shift verso sinistra equivale a moltiplicare il dato per due.

SRL m

Quest'operazione effettua lo shift a destra del dato m: ogni bit viene spostato al posto del bit che si trova alla sua destra. Al posto del bit sette viene inserito uno zero. Il vecchio contenuto del bit sette viene posto nel flag di carry.



Modifica del registro dei flag

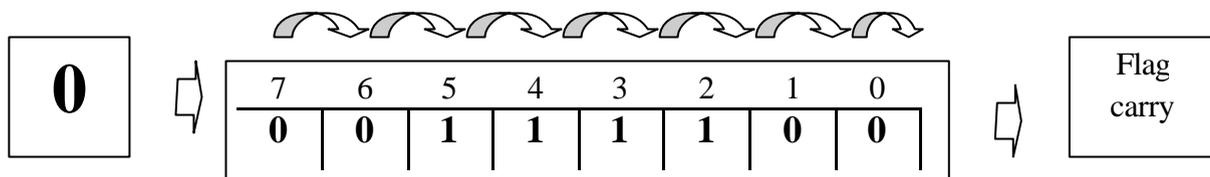


- Il flag S funziona come flag di segno, e rappresenta il segno del dato dopo lo shift
- Il flag Z funziona come flag di zero
- H viene posto sempre a zero
- Il flag PV funziona come flag di parità
- Il flag N è posto sempre a zero
- Il flag di carry conterrà il bit sette del dato

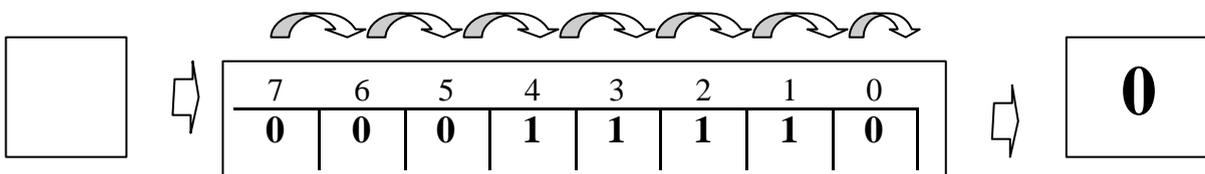
✓ L'operando m può essere un registro ad otto bit dello Z80

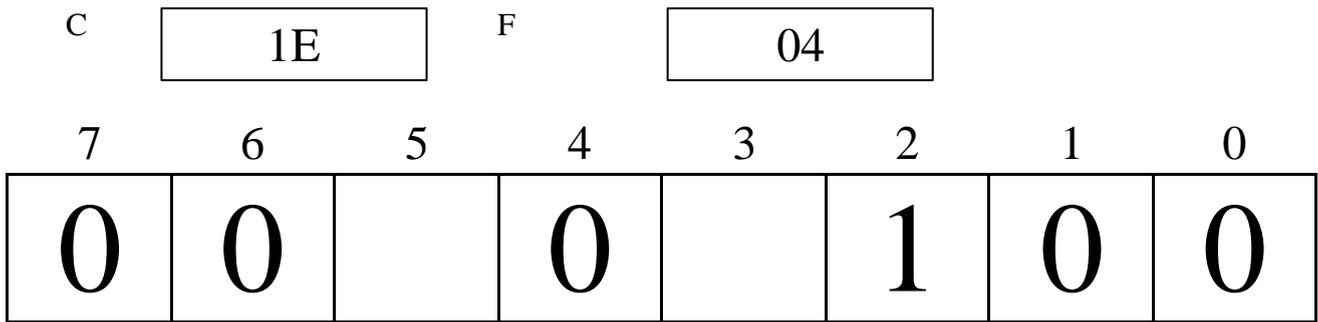
Es. SRL C

Prima



Dopo



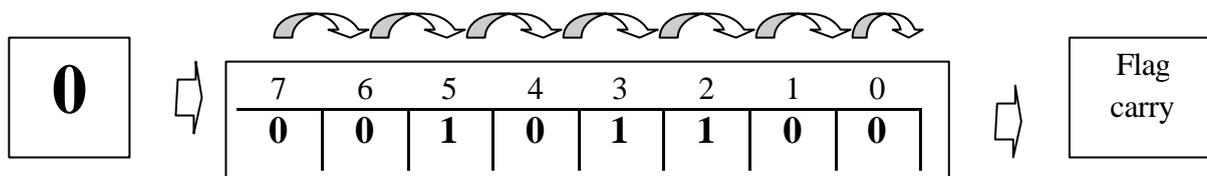


- Il risultato dello shift è stato un numero positivo (S=0)
 - Il risultato dello shift è stato un numero diverso da zero (Z=0)
 - Il risultato dello shift presenta un numero di bit ad uno pari a quattro (numero pari) per cui PV=1
 - Il bit sette del dato prima dello shift era zero per cui C = 0
 - ✓ L'operando m può essere un operando di memoria il cui indirizzo si trova nel registro HL (indirizzato in modo indiretto)
- Es. SRL (HL)

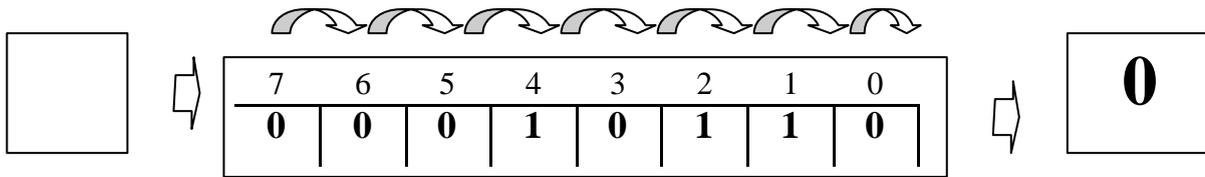
Prima



1C00	2C
1C01	3B
1C02	41
1C03	56

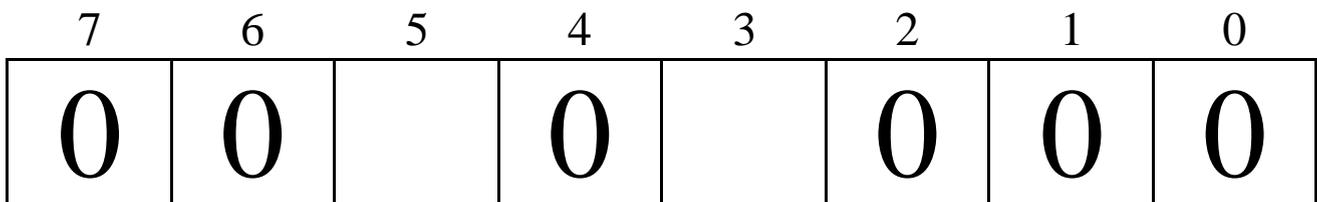


Dopo



HL 1C00 F 00

1C00	16
1C01	3B
1C02	41
1C03	56



- Il risultato dello shift è stato un numero positivo (S=0)
- Il risultato dello shift è stato un numero diverso da zero (Z=0)
- Il risultato dello shift presenta un numero di bit ad uno pari a tre (numero pari) per cui PV=0
- Il bit zero del dato prima dello shift era zero per cui C = 0

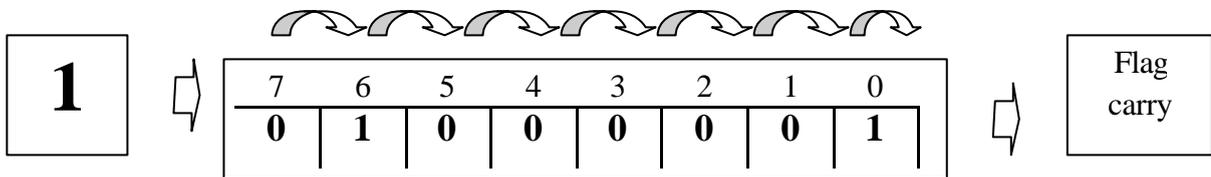
✓ L'operando m può essere il contenuto di una locazione di memoria il cui indirizzo si ottiene sommando il contenuto di un registro indice (IX o IY) ad uno spiazzamento (indirizzamento indicizzato)

Es. SRL (IX+2)

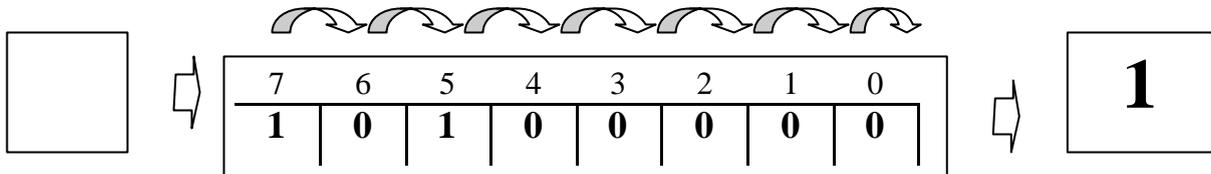
Prima

IX 1C00 F 11

1C00	2C
1C01	3B
1C02	41
1C03	56



Dopo



IX 1C00 F 85

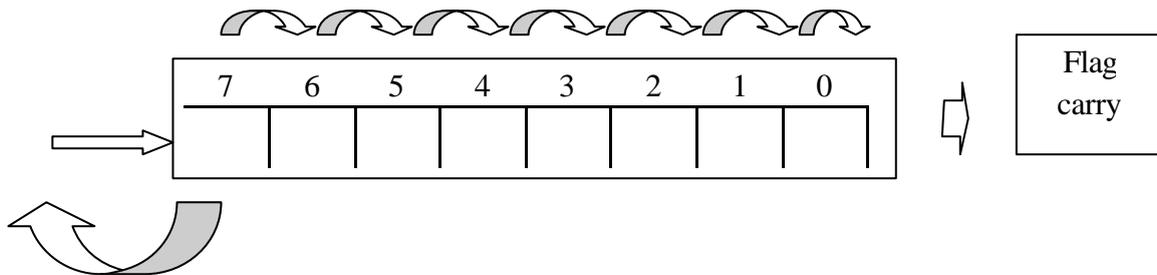
1C00	96
1C01	3B
1C02	A0
1C03	56

7	6	5	4	3	2	1	0
1	0		0		1	0	1

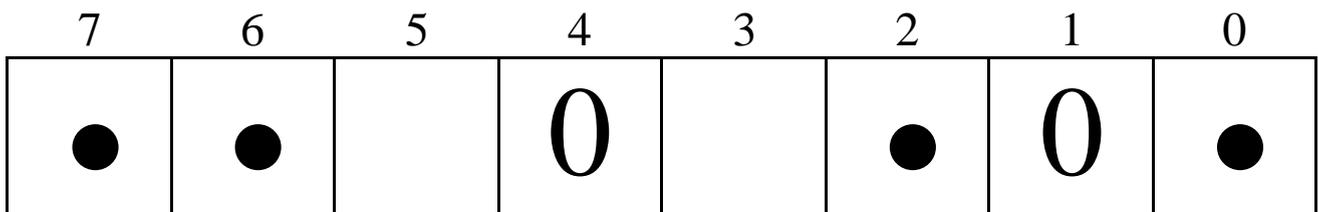
- Il risultato dello shift è stato un numero negativo (S=1)
- Il risultato dello shift è stato un numero diverso da zero (Z=0)
- Il risultato dello shift presenta un numero di bit ad uno pari a due (numero pari) per cui PV=1
- Il bit zero del dato prima dello shift era uno per cui C = 1

SRA m

Quest'istruzione effettua anch'essa lo shift verso destra del dato. Vi è, però, una differenza con lo shift a destra logico. Nello shift aritmetico lo spostamento verso destra di ogni bit equivale a dividere il dato che si vuole shiftare per due. Se, però, si vuole dare un significato matematico all'operazione il dato non deve cambiare di segno. Il segno del dato è espresso dal bit sette per cui, quando si effettua lo shift, il bit sette viene spostato verso destra ma va contemporaneamente a rioccupare il proprio posto. In questo modo il segno del dato si conserva anche dopo l'operazione di shift.



Modifica del registro dei flag

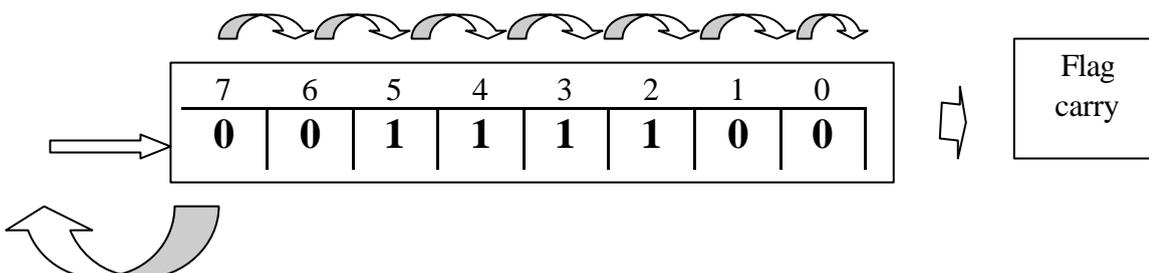


- Il flag S funziona come flag di segno, e rappresenta il segno del dato dopo lo shift
- Il flag Z funziona come flag di zero
- H viene posto sempre a zero
- Il flag PV funziona come flag di parità
- Il flag N è posto sempre a zero
- Il flag di carry conterrà il bit sette del dato

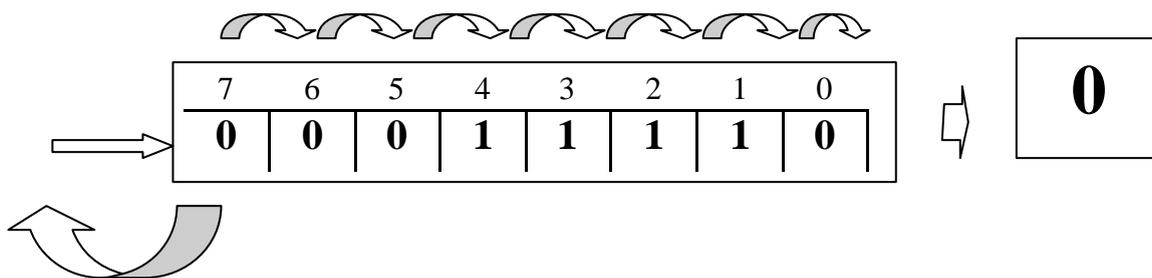
✓ L'operando m può essere un registro ad otto bit dello Z80

Es. SRA C

Prima



Dopo



C 1E F 04

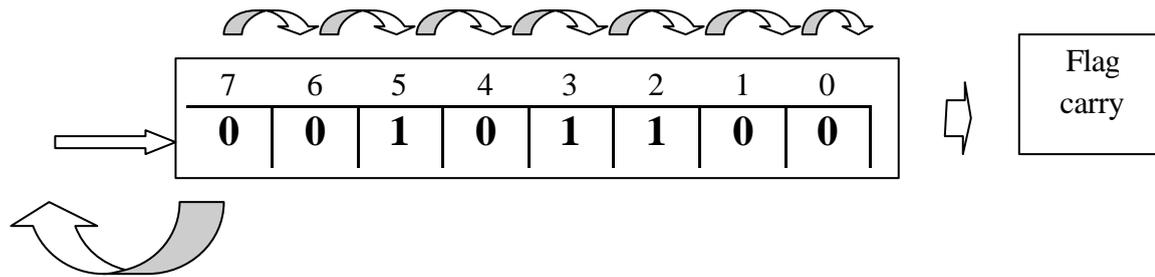
7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	0

- Il risultato dello shift è stato un numero positivo (S=0)
 - Il risultato dello shift è stato un numero diverso da zero (Z=0)
 - Il risultato dello shift presenta un numero di bit ad uno pari a quattro (numero pari) per cui PV=1
 - Il bit sette del dato prima dello shift era zero per cui C = 0
 - ✓ L'operando m può essere un operando di memoria il cui indirizzo si trova nel registro HL (indirizzato in modo indiretto)
- Es. SRA (HL)

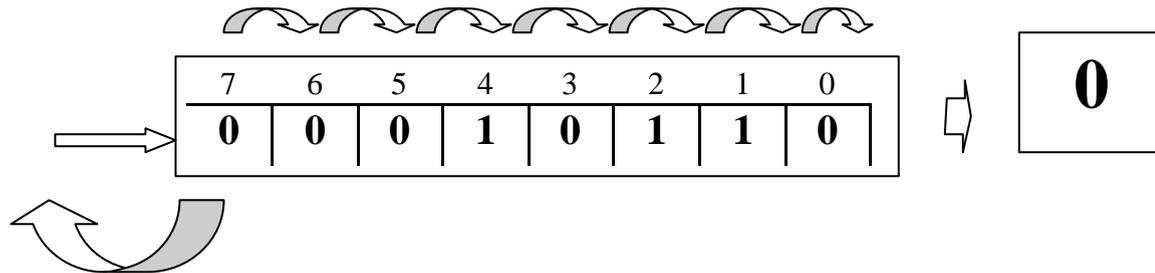
Prima

HL 1C00 F 10

1C00	2C
1C01	3B
1C02	41
1C03	56



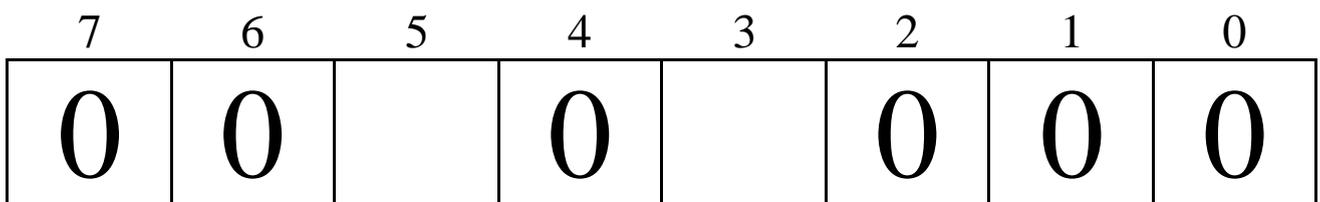
Dopo



HL 1C00

F 00

1C00	16
1C01	3B
1C02	41
1C03	56

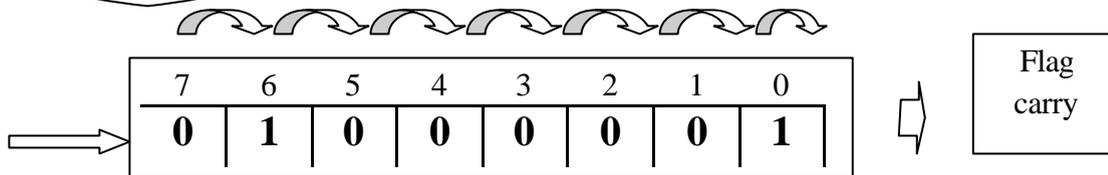
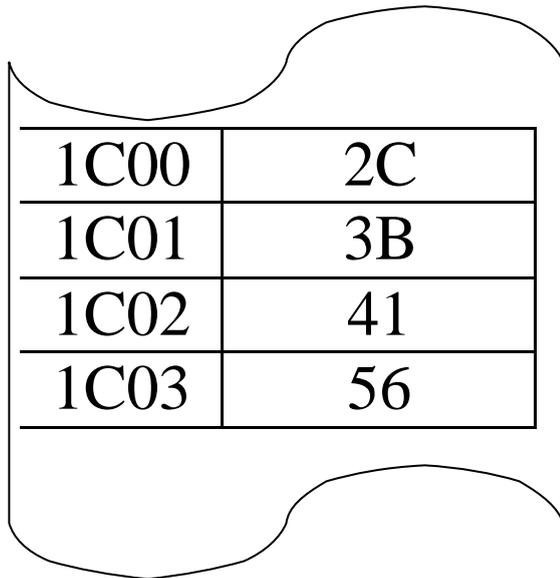


- Il risultato dello shift è stato un numero positivo (S=0)
- Il risultato dello shift è stato un numero diverso da zero (Z=0)
- Il risultato dello shift presenta un numero di bit ad uno pari a tre (numero pari) per cui PV=0

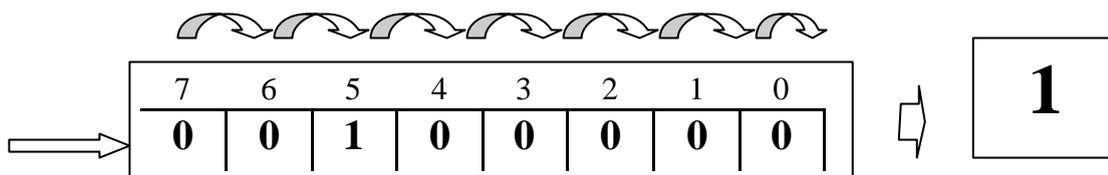
- Il bit zero del dato prima dello shift era zero per cui $C = 0$
 - ✓ L'operando m può essere il contenuto di una locazione di memoria il cui indirizzo si ottiene sommando il contenuto di un registro indice (IX o IY) ad uno spiazzamento (indirizzamento indicizzato)

Es. SRA (IX+2)

Prima



Dopo



1C00	2C
1C01	2B
1C02	20
1C03	56

7	6	5	4	3	2	1	0
0	0		0		0	0	0

- Il risultato dello shift è stato un numero positivo ($S=0$)
- Il risultato dello shift è stato un numero diverso da zero ($Z=0$)
- Il risultato dello shift presenta un numero di bit ad uno pari ad uno (numero dispari) per cui $PV=0$
- Il bit zero del dato prima dello shift era zero per cui $C = 0$

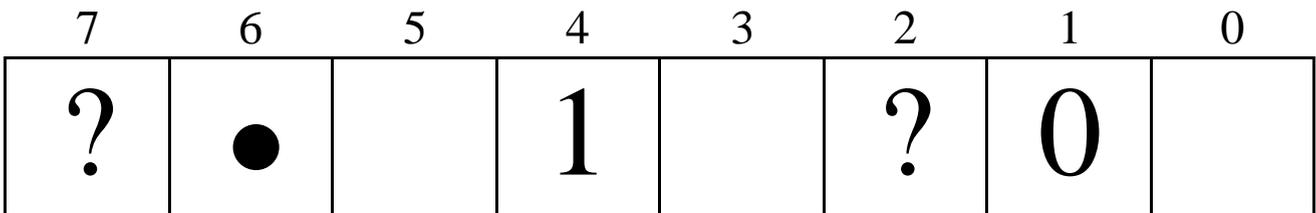
ISTRUZIONI DI BIT, SET, RESET

BIT b, m

Quest'istruzione ha lo scopo di controllare il bit numero b dell'operando m. il risultato del test sarà posto nel flag Z secondo la seguente legge

- ✓ Se il bit testato risulta pari ad 1, il flag di zero viene posto pari a 0
- ✓ Se il bit testato risulta pari a 0, il flag di zero viene posto pari a 1

Modifica del registro dei flag

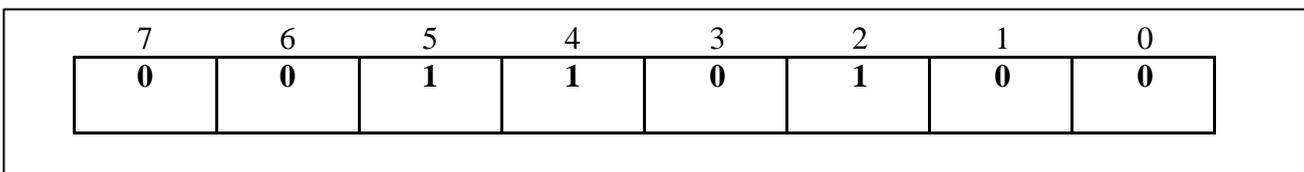


- Il flag S assume un valore casuale
- Il flag Z funziona nella modalità descritta sopra
- H viene posto sempre ad uno
- Il flag PV assume un valore casuale
- Il flag N è posto sempre a zero
- Il flag di carry non viene influenzato dall'operazione e conserva il suo vecchio valore

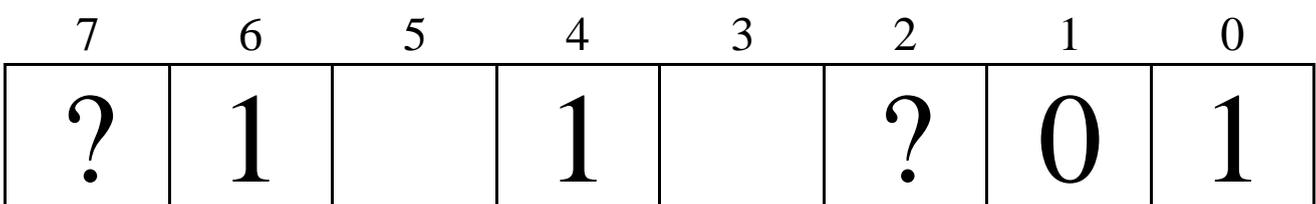
- ✓ L'operando m può essere un registro ad otto bit del microprocessore

Es. Bit 3, D

Prima

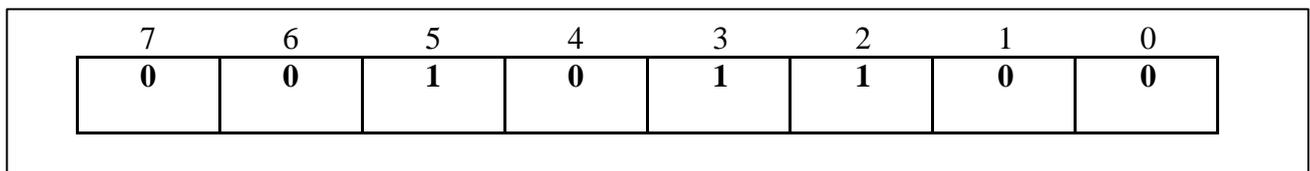
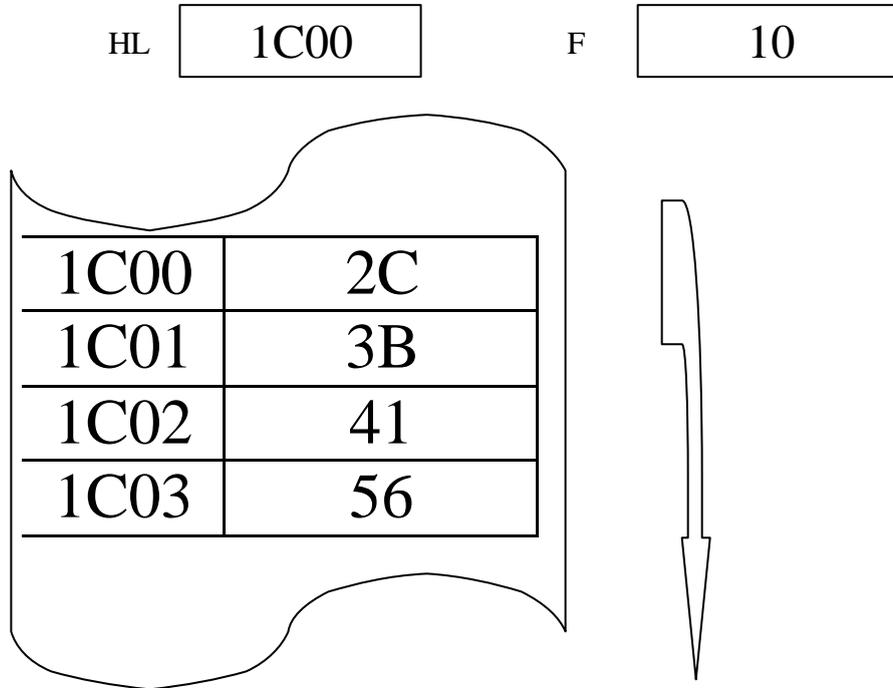


Il bit 3 dell'operando è pari a zero per cui il flag di zero verrà posto ad 1

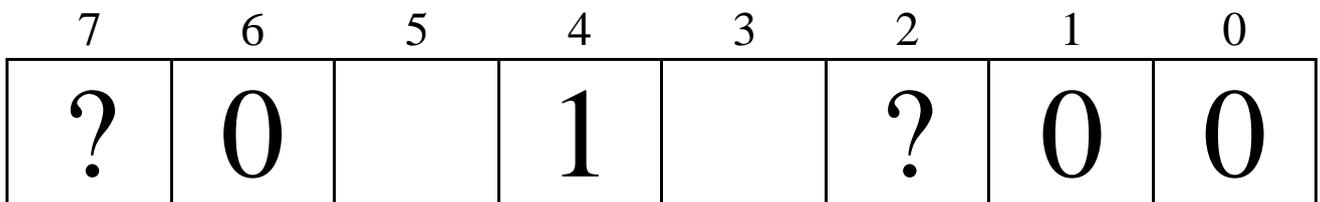


Il flag di carry è ad 1 poiché tale era il suo valore prima.

- ✓ L'operando s può essere una locazione di memoria indirizzata in modo indiretto
- Es. Bit 5, (HL)



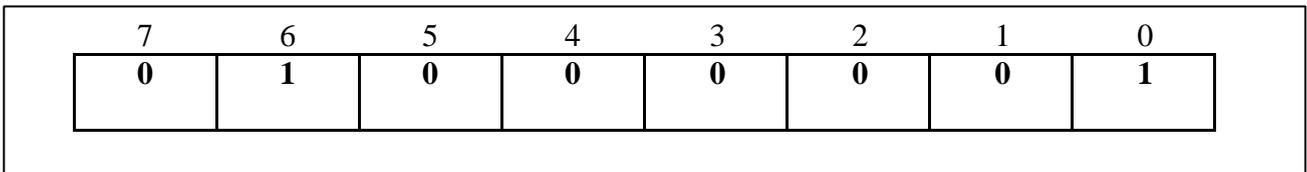
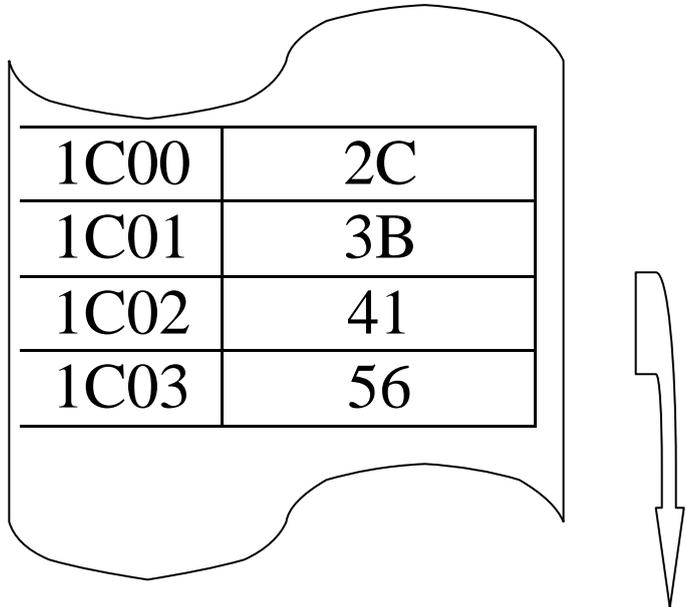
Il bit 5 dell'operando è pari ad uno per cui il flag di zero verrà posto ad 0



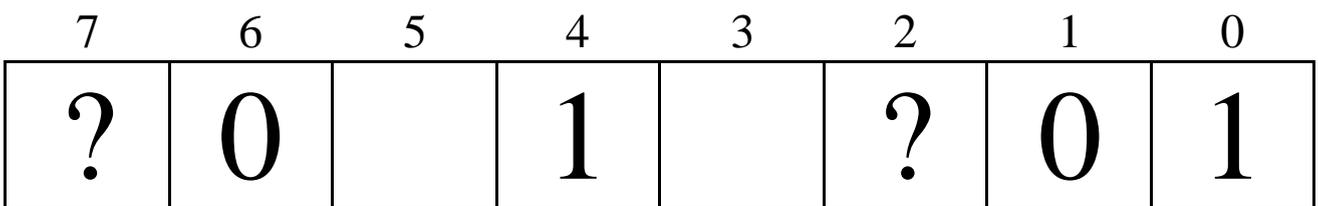
✓
✓

✓ L'operando m può essere una locazione di memoria indirizzata in modo indicizzato

Es, Bit 7, (IX+2)



Il bit 7 dell'operando è pari a zero per cui il flag di zero verrà posto ad 1



SET b, m

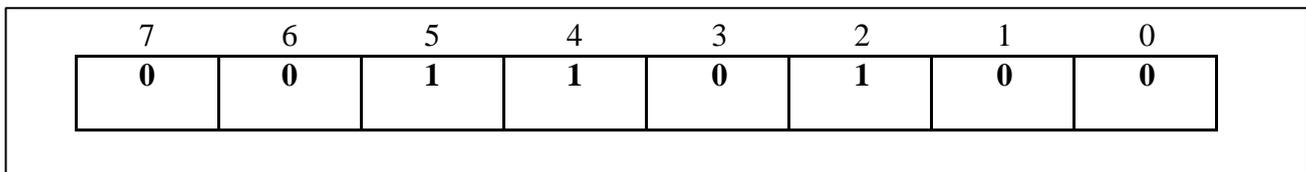
Quest'istruzione ha lo scopo di porre ad 1 il bit b dell'operando s

- ✓ L'operando m può essere un registro ad otto bit del microprocessore

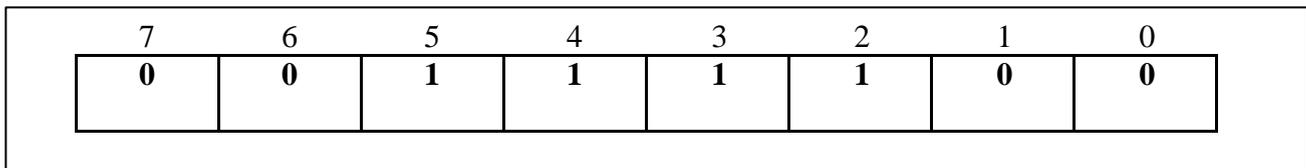
Es. Set 3, D

Prima

D 34



Dopo



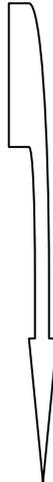
D 3C

- ✓ L'operando s può essere una locazione di memoria indirizzata in modo indiretto

Es. Set 5, (HL)

HL 1C00

1C00	1C
1C01	3B
1C02	41
1C03	56



7	6	5	4	3	2	1	0
0	0	0	1	1	1	0	0

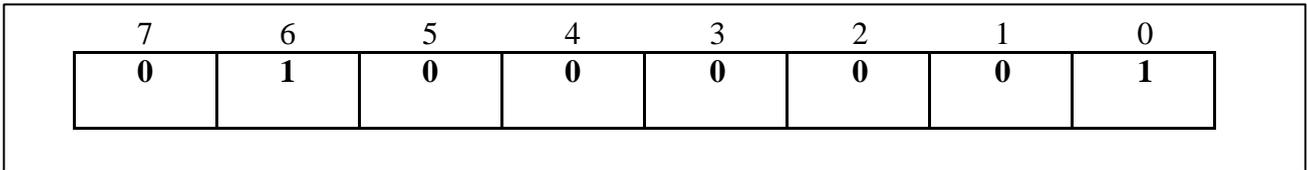
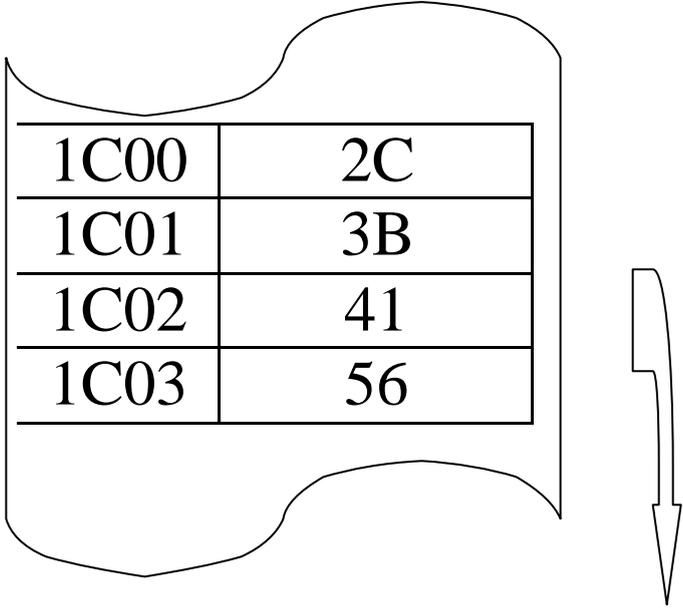
Dopo

7	6	5	4	3	2	1	0
0	0	1	1	1	1	0	0

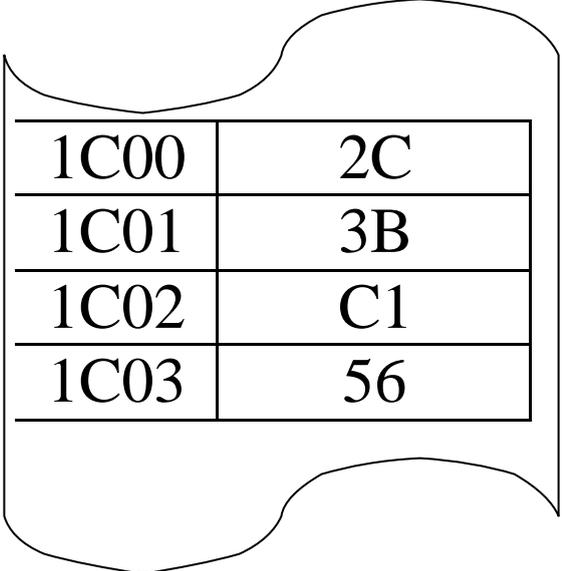
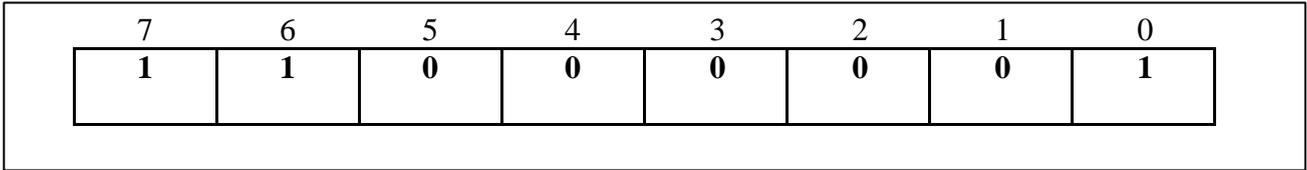
1C00	3C
1C01	3B
1C02	41
1C03	56

✓ L'operando m può essere una locazione di memoria indirizzata in modo indicizzato

Es, Set 7, (IX+2)



Dopo



RES b, m

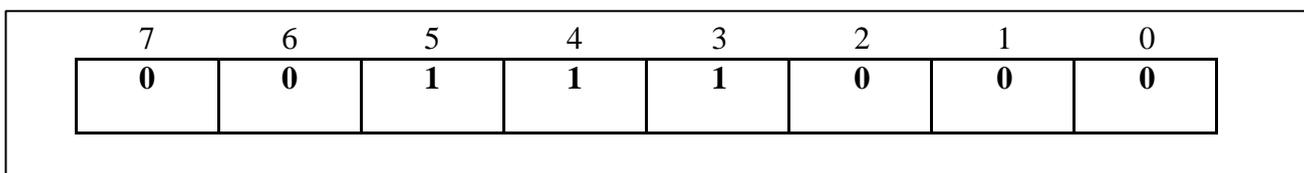
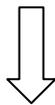
Quest'istruzione ha lo scopo di porre a 0 il bit b dell'operando s

- ✓ L'operando m può essere un registro ad otto bit del microprocessore

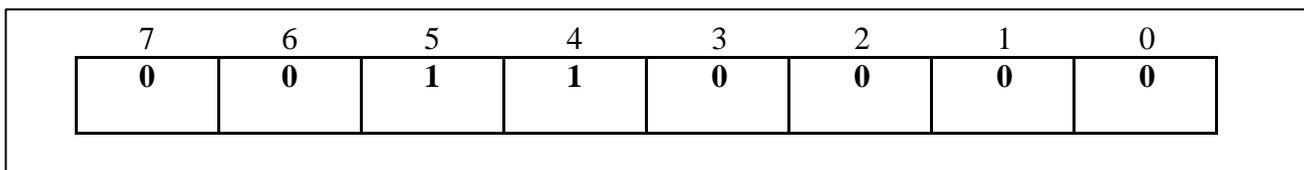
Es. Res 3, D

Prima

D 38



Dopo



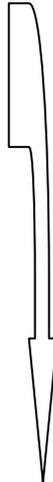
D 30

- ✓ L'operando s può essere una locazione di memoria indirizzata in modo indiretto

Es. Res 5, (HL)

HL 1C00

1C00	3C
1C01	3B
1C02	41
1C03	56



7	6	5	4	3	2	1	0
0	0	1	1	1	1	0	0

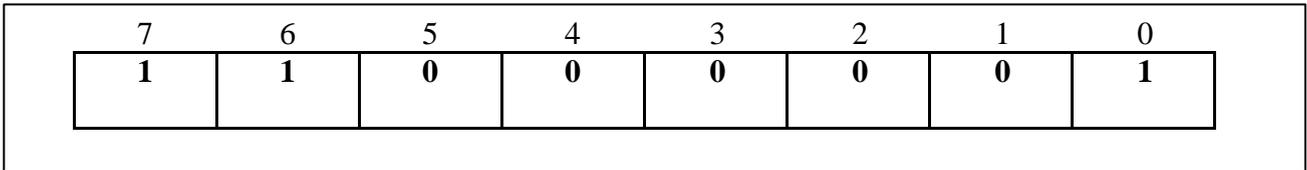
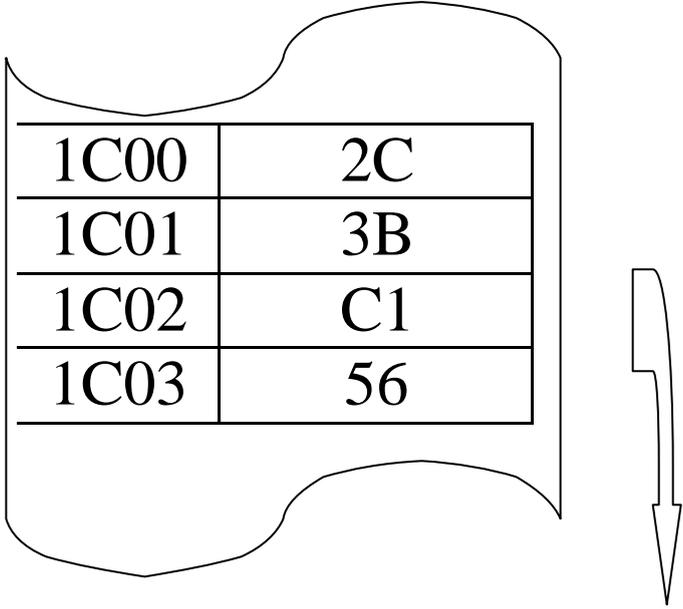
Dopo

7	6	5	4	3	2	1	0
0	0	0	1	1	1	0	0

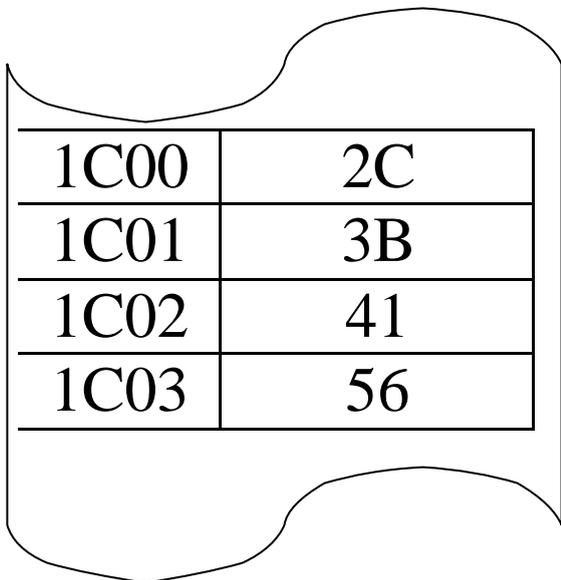
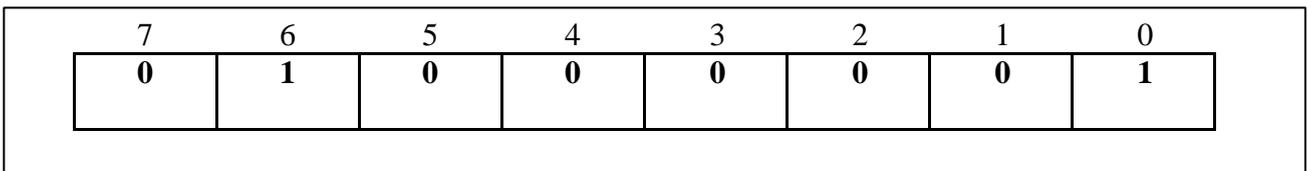
1C00	1C
1C01	3B
1C02	41
1C03	56

✓ L'operando m può essere una locazione di memoria indirizzata in modo indicizzato

Es, Res 7, (IX+2)



Dopo



ISTRUZIONI DI SALTO**JP pq**

E' un'istruzione di salto assoluto incondizionato. Per salto assoluto s'intende il fatto che grazie a quest'istruzione, il contenuto del Program Counter viene sostituito completamente con il nuovo indirizzo pq espresso direttamente all'interno dell'istruzione. In tal modo si può saltare ad una qualunque locazione di memoria. Si dice incondizionato poiché questo salto viene eseguito senza dover sottostare a nessuna condizione. In sostanza il microprocessore è obbligato ad eseguire il salto quando incontra l'istruzione durante l'esecuzione del programma. L'istruzione non modifica il registro dei flag.

Es. JP 3A52

Prima

PC

1B00

Dopo

PC

3A52

JP cc, pq

E' un'istruzione di salto assoluto condizionato. Quest'istruzione ha un effetto identico a quello dell'istruzione precedente ma a differenza di essa viene eseguita soltanto se la condizione cc espressa nell'istruzione risulta essere vera. Le condizioni sono espresse nella seguente tabella

JP NC, pq	Esegui il salto se il flag di carry si trova a zero
JP C, pq	Esegui il salto se il flag di carry si trova ad uno
JP NZ, pq	Esegui il salto se il flag di zero si trova a zero
JP Z, pq	Esegui il salto se il flag di zero si trova ad uno

