

ESERCIZIO 1 TER	1
Traccia	1
Analisi	1
Codifica programma	6

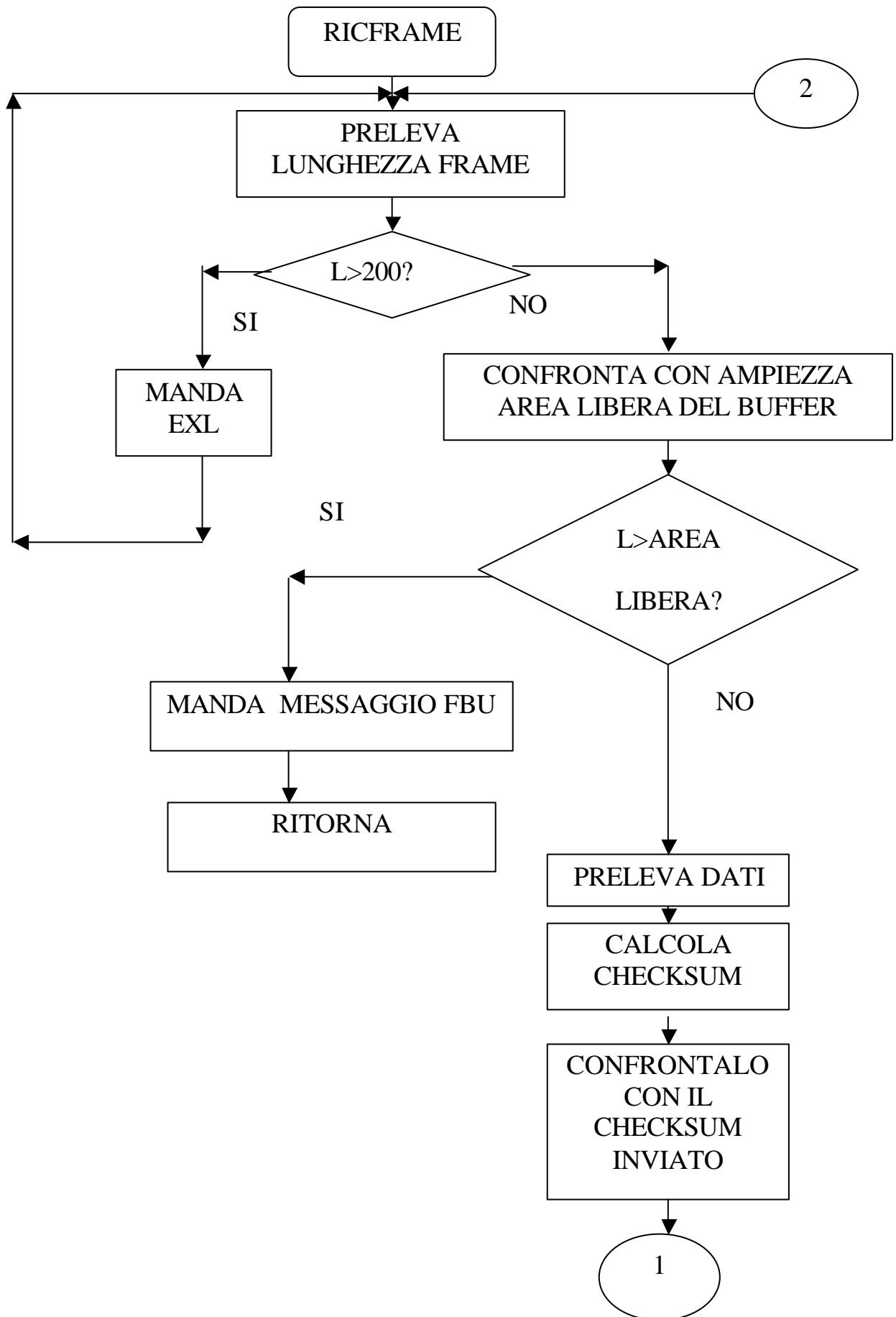
Esercizio 1 ter

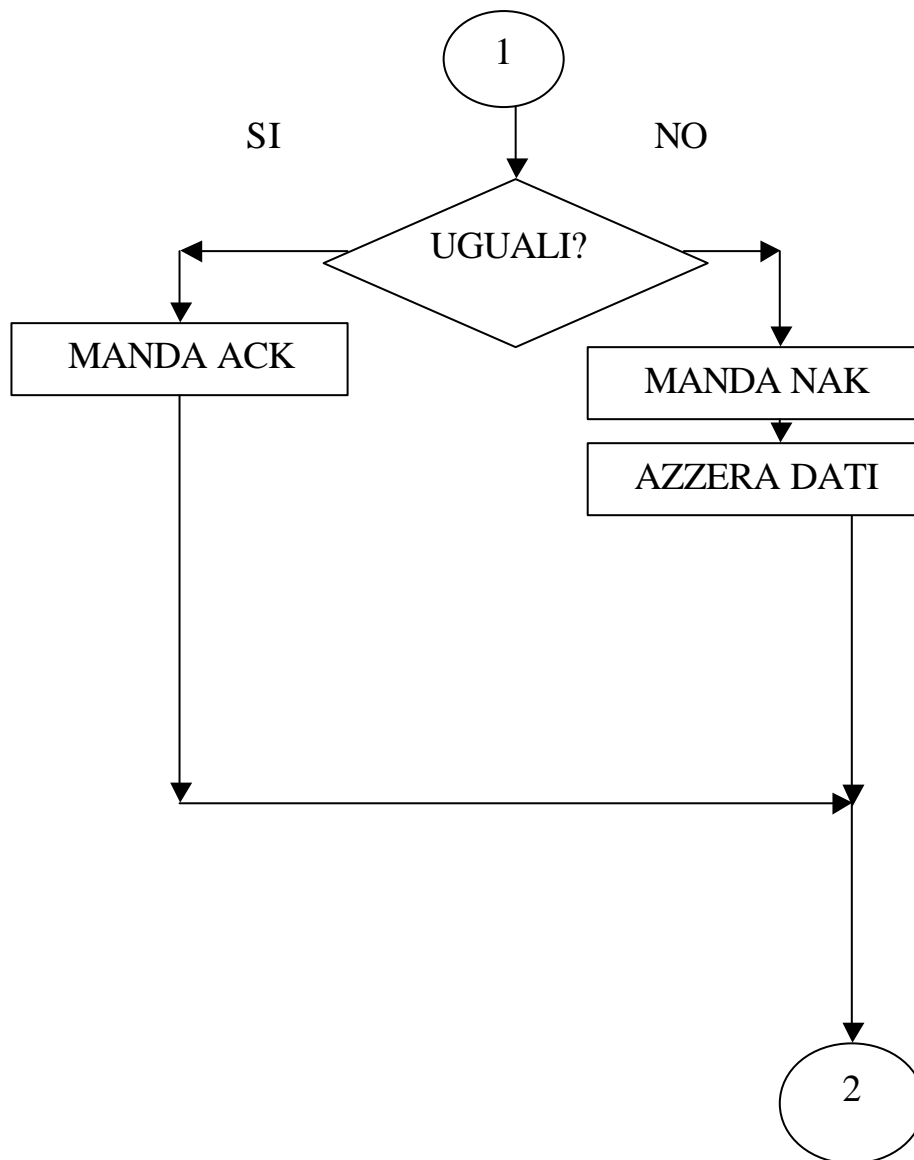
Traccia

[Vedi Esercizio 1](#)

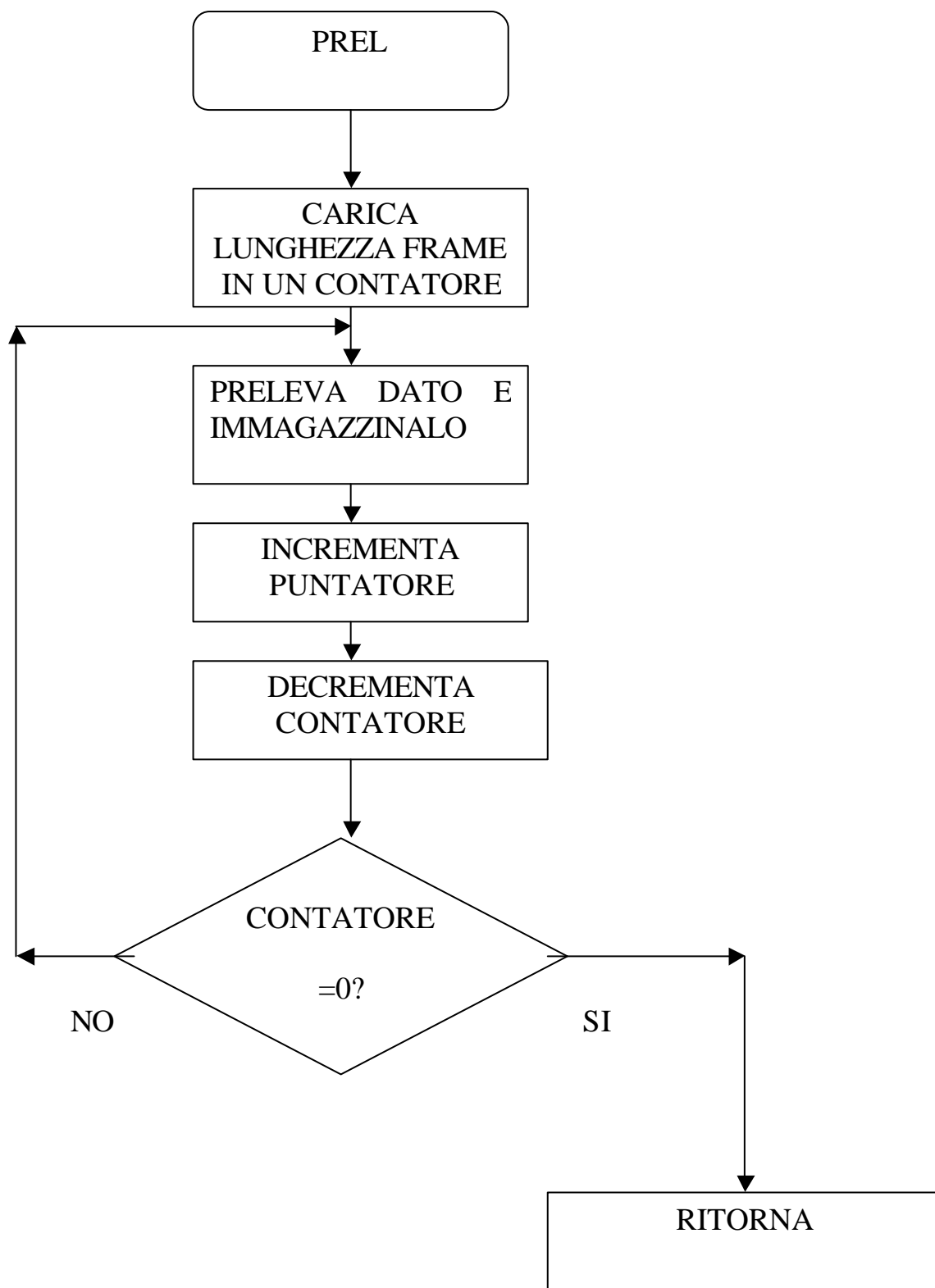
Analisi

L'analisi per l'esercizio, fatta precedentemente non è perfetta. Si noti, infatti, che, almeno a quanto traspare dalla traccia, i frame possono avere lunghezza variabile ancorché minore o uguale a 200. Ora, l'effettuazione di un controllo sullo stato di riempimento del buffer, fatto al termine di ogni singolo frame funziona solo se i frame sono tutti di ampiezza pari esattamente a 200 byte. Se così non fosse si potrebbe avere il caso, ad esempio, che il penultimo frame immesso abbia portato l'area occupata del buffer a 9920 locazioni, per cui restano soltanto 80 locazioni libere e arriva un nuovo frame di, ad esempi, 110 byte. Con il programma visto prima, il microprocessore riterrebbe che il buffer è ancora vuoto, ma la scrittura del frame nuovo deborderebbe dal buffer. Risolviamo il problema effettuando all'inizio il controllo confrontando l'area libera del buffer con la lunghezza del frame.

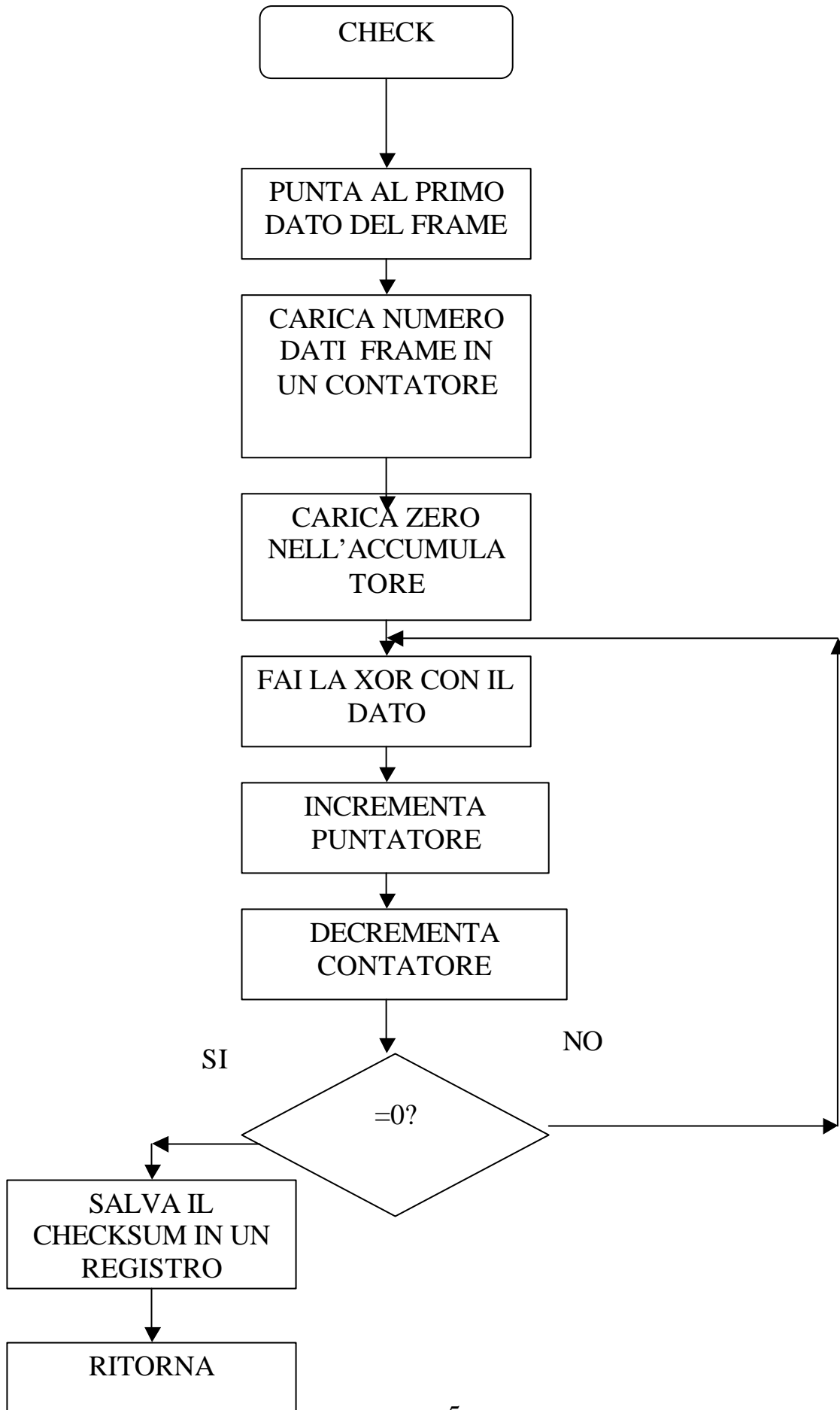




Il blocco preleva dati può essere realizzato con un altro sottoprogramma PREL che può essere rappresentato dal seguente diagramma



Il blocco calcola checksum può essere realizzato con un altro sottoprogramma CHECK



Codifica programma

RICFRAME: LD HL,1A00H; punta all'inizio del buffer

ET: IN A,(30h); leggi la lunghezza del frame contenuta nel primo byte

CP C9h;controlla se la lunghezza è superiore a 200, pari a C8 hex

JP N,ET1;se la lunghezza del frame è inferiore salta le istruzioni

;che inviano il messaggio EXL

LD A,A3h; carica in A il codice di EXL

OUT (30h),A;invialo alla porta

JP ET;torna all'inizio

ET1: LD B,00h

LD C ,A ;poniamo in BC la lunghezza del frame

PUSH HL;salva HL nello stack

ADD HL,BC; controlla a che indirizzo si arriverebbe inserendo il nuovo

;frame

AND A; azzera il flag di carry

SBC HL,4110h;è l'indirizzo della prima locazione successiva al buffer

; se posso inserire il frame HL dovrà arrivare al massimo a questa

;locazione, in caso contrario avremmo lo sfioramento dell'area buffer

;e la sottrazione darebbe esito negativo

JP P, ET5 ; se è tutto a posto preleva il frame

LD A,FEh; altrimenti invia il messaggio di buffer pieno

OUT(30h),A

RET; ed esci dal sottoprogramma

ET5: ADD HL,4110h

AND A ;

SBC HL,BC ;ripristina HL dopo le operazioni di controllo

CALL PREL; se la lunghezza è giusta preleva i dati,

 ;il sottoprogramma deve lasciare A inalterato poiché

 ; la lunghezza del frame

CALL CHECK;al termine del sottoprogramma HL punterà

 ; alla locazione che contiene il checksum inviato,

 ; imponiamo che lasci in D il checksum

 ;calcolato

LD E,A;salva A che contiene la lunghezza del frame

LD A,D;poni in A il checksum calcolato

CP (HL);confronta con il checksum calcolato

JP Z,ET2;se tutto è a posto invia ACK

LD A,FEh;altrimenti inserisci in A il valore di NAK

OUT(30h),A

LD A,E; recupera la lunghezza del frame

CALL AZZERA

JP ET;si ritorna all'inizio attendendo l'invio di un checksum corretto

ET2: LD A,FFh ;carica in A il codice ACK

OUT(30h),A

JP ET; torna all'inizio

PREL: LD B,A;salva nel contatore B la lunghezza del frame

; HL contiene già l'indirizzo della prima locazione del buffer

: da riempire

INIR

RET

CHECK: LD C,A

LD B,00h;abbiamo inserito nel registro BC la lunghezza del frame

AND A;azzerà il carry

SBC HL,BC;il sottoprogramma PREL ha portato HL più avanti di una

; posizione per cui se sottraggo la lunghezza del frame

;mi porto al primo byte del frame

INC HL;mi devo portare al secondo byte che è il primo dei dati

LD B,C;uso B come contatore

DEC B;deve essere pari alla lunghezza del frame meno uno

;poiché non dovremo fare la XOR anche con il byte di checksum

;inviato dal trasmettitore

LD E,A;salvo in E la lunghezza del frame

LD A,00h;A conterrà il checksum

ET4: XOR (HL)

INC HL

DJNZ ET4

LD D,A ;salva il checksum calcolato

LD A,E;salva di nuovo in A la lunghezza del frame

RET

AZZERA: LD C,A

;il sottoprogramma CHECK ha fatto in modo che HL punti

;all'ultimo byte del frame, ad esempio se si trattava del primo pacchetto e questo era

;composto di 100 dati il buffer era stato riempito da 1A00H a 1A63H, per cui dopo

;CHECK il registro HL contiene 1A63H

;per tornare indietro basta sottrarre ad HL la lunghezza del frame (64 in esadecimale)

;meno 1

DEC C; lunghezza del frame meno uno

LD B,00h ; facciamo in modo che BC contenga la lunghezza del frame

;con la parte alta a zero

AND A ; azzeriamo il flag di carry poiché dobbiamo usare SBC

SBC HL,BC

RET