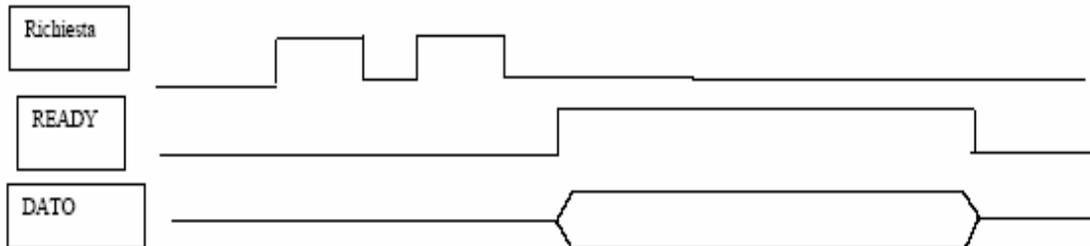


Esercizio 18.1

Scrivere un sottoprogramma che acquisisca da una periferica di input una linea di caratteri con al più N dati e li memorizzi in memoria a partire da un indirizzo assegnato.

Il sottoprogramma accetta come parametri di ingresso il numero max. di caratteri (N), indirizzo iniziale dell'area di memoria e un carattere terminatore. Il driver termina la lettura o se viene letto il carattere terminale (passato come parametro) o se sono stati acquisiti N caratteri. Il driver restituisce al programma chiamante il numero di caratteri effettivamente letti. Il protocollo utilizzato per la lettura del singolo carattere è quello specificato dal diagramma riportato in figura. Il registro di controllo e stato è mappato in memoria all'indirizzo \$8500 e il registro dato all'indirizzo \$8501. Il bit di richiesta e il bit ready sono rispettivamente il bit 2 e il bit 5 del registro di controllo e stato.



```

                                org $8000
RCS          equ          $8500
RDATI       equ          $8501
tappo       equ          '*'
N           equ          10
mem         ds.b         10
letti       ds.b         1
start       movem.l      d0/d1/d2/a0, -(SP)
            move.w       #N,d0
            move.b       #tappo,d1
            move.l       #mem,a0
            jsr          leggi
            move.b       d2,letti
            movem       (SP)+, d0/d1/d2/a0
            stop        #$2700
leggi       move.l       d4,-(SP)
            clr.b       d2
loop        btst.b      #5,rcs          attesa su ready
            bne         loop
            bset.b      #2,RCS         avvia il protocollo
            bclr.b     #2,RCS
            bset.b      #2,RCS
            bclr.b     #2,RCS
wait       btst.b      #5,RCS         attesa attiva su ready
            beq         wait
            move.b      RDATI,d4      lettura del carattere
            move.b      d4,(a0)+
            add.b       #1,d2
            cmp.b       d4,d1
            dbeq        d0,loop
            movem.l    (SP)+,d4
            rts

```

end

start