A7 = Char Addr
A2 = Req Length
A3 = Eff Length
A4 = Order (4 LSB)

Φ:TPUT

entry: { A6 = DWT
CF A15, Φ:TPUT

ML 4 ⇐ 12+A6
A5 ⇐ 0
A4 - 5
A4 * 2

ABI SWFUNO, A4

order = 5 | =6 | =7 | =8

BINΦUT

=A3 A2 ≠A3

BINΦT2

E:NDIΦ

BINΦT1

[A4]R ⇐ (A7)

BINΦT4

A7 + 7
A3 + 7

BINΦT5

MS 3 ⇒ 12+A6

BINΦT3

word to output

A4 ⇒ 22+A6

3

RTN A15

ASCΦUT

<A2 A3 ≥A2

A4 ⇐ (A6)

='TP' A4 ≠'TP'

ASRTPO

insert spaces before type off

A2 ⇐ (26+A6)
A2 - 5

≥0 A2 <0

ASCOO1

A2 + 6
[A4]R ⇐ (ASCBUF+A2)
26+A6+7

A4 ⇐ (A6)

='TY' A4 ≠'TY'

ASCOOO

A4 ⇐ 1700A
!!!
SVK A5,10

output 5 char of ASCBUF

ΦBJΦT4

4 * 4 obj output

ΦBJΦT8

[A4]R ⇐ (A7)

=1 A3 ≤0

>1

ΦBJΦ87

conversion of the 1st char

BINΦT5

≠A3 A2 =A3

check word → A4 ⊕ 5 ⇒ 24+A6

A4 ⇐ 1700
!!!
SVK A5,6

A4 ⇒ ΦBJΦ872

ΦBJΦ83

A4 ⇐ (24+A6)
A5 ⇐ (26+A6)
26+A6+7

=0 A5 ≠0

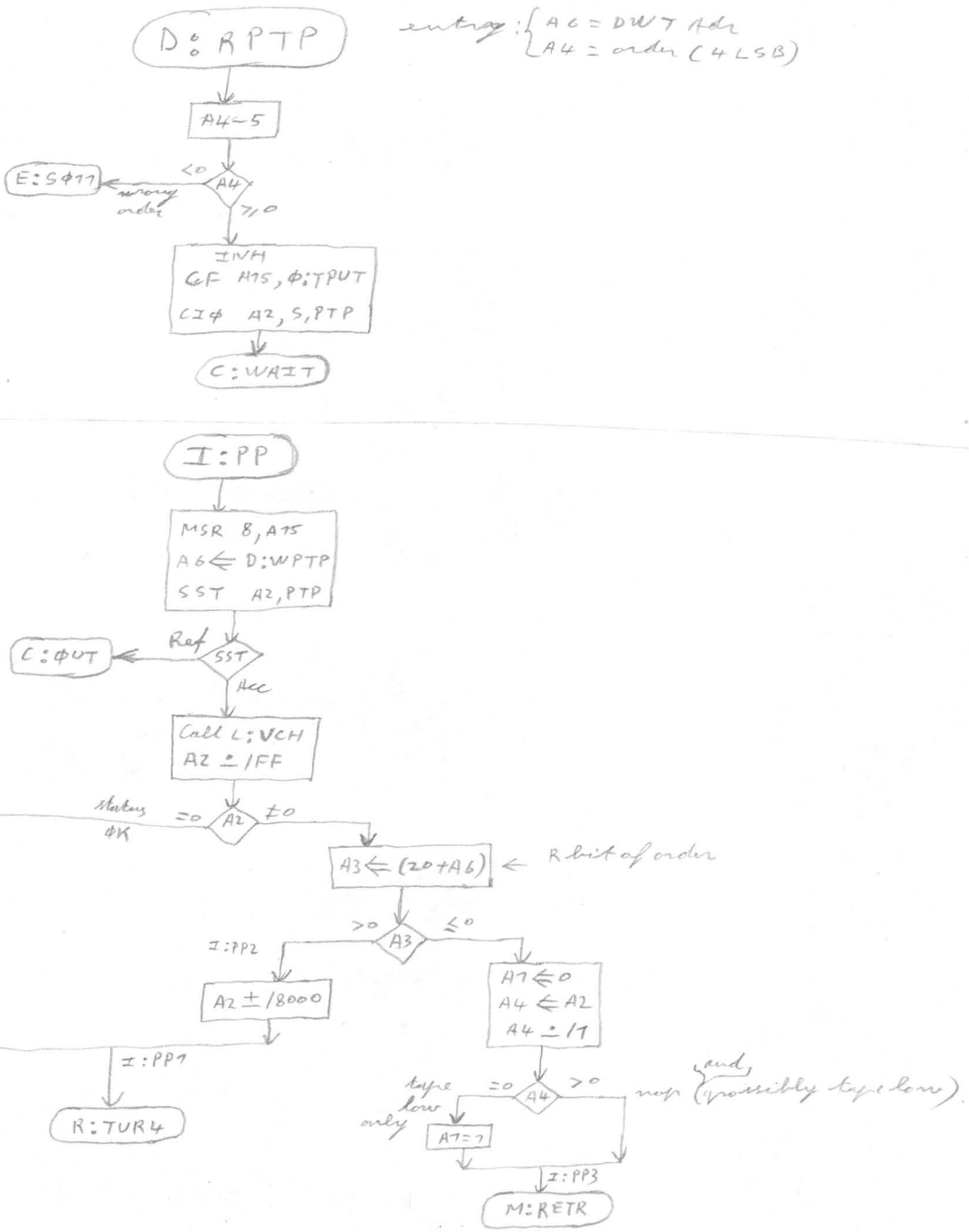
ΦBJΦ87

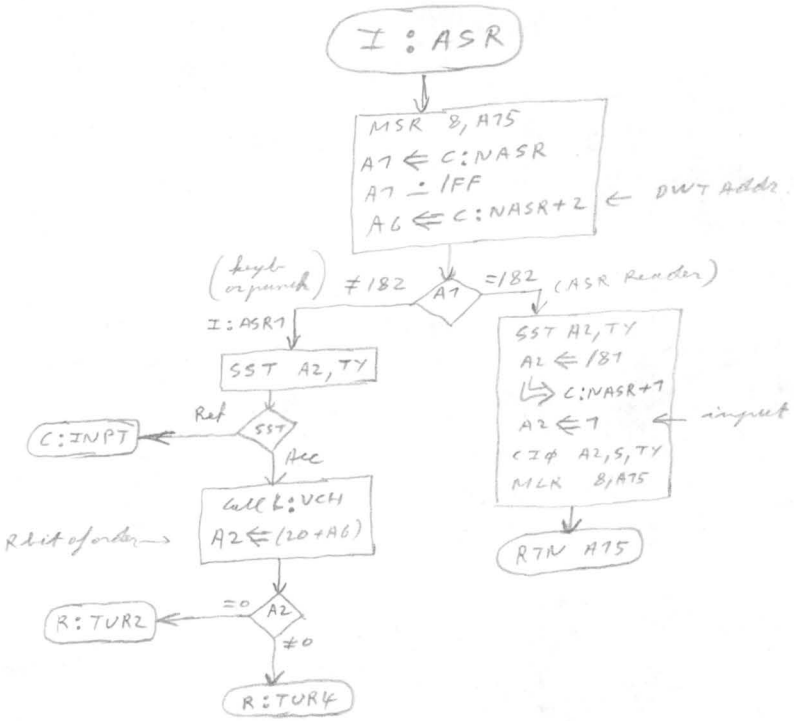
A4 ⇐ 0
A5-10 or A5-6

≠0 A5 =0

A4 ⇒ 24, A6

output 5 NUL chars.


D:RPTP
entry: { AC = DWT Adr
A4 = order (4 LSB)
A4←5
<0
A4
E:SØTT
wrong order
≥0
INH
GF A15, Ø:TPUT
CIØ A2, S, PTP
C:WAIT
I:PP
MSR B, A15
A6 ⇐ D:WPTP
SST A2, PTP
Ref
SST
C:ØUT
Acc
Call L:VCH
A2 ± 1FF
Status ØK
=0
A2
≠0
I:PP7
A3 ⇐ (20+A6)
R bit of order
>0
A3
≤0
I:PP2
A2 ± 18000
A7 ⇐ 0
A4 ⇐ A2
A4 ± 17
I:PP7
R:TUR4
type low only
=0
A4
>0
nop (and possibly type low)
A7=7
I:PP3
M:RETR


I : ASR
MSR 8, A75
A7 ⇐ C:NASR
A7 ∸ 1FF
A6 ⇐ C:NASR+2
← DWT Addr.
(keyb. request)
≠182
A7
=182
(ASR Reader)
I:ASR7
SST A2, TY
Ref
SST
C:INPT
Acc
Call L:VCH
A2 ⇐ (20+A6)
R bit of order →
=0
A2
R:TURZ
≠0
R:TURY
SST A2, TY
A2 ⇐ 181
⇨ C:NASR+7
A2 ⇐ 7
← input
CIO A2, 5, TY
MLR 8, A75
RTN A75

D:RAS1

entry: { A6: DWT Addr
A4: order (4 LSB)

A5 ⇐ /81
[A5]R ⇒ C:NASR+1
A6 ⇒ C:NASR+2
A4 - 5

A4 <0 (input) DRASIN
A4 =0 (basic write)
A4 >0 (standard write)

DRASIN:
A2 ⇐ 1 ← input
CIφ A2,5,TY
→ C:WAIT

A4 ⇐ (12, A6) ← Buff Addr

A4 ≠E:SECB / =E:SECB
A4 ≠E:FECB / =E:FECB

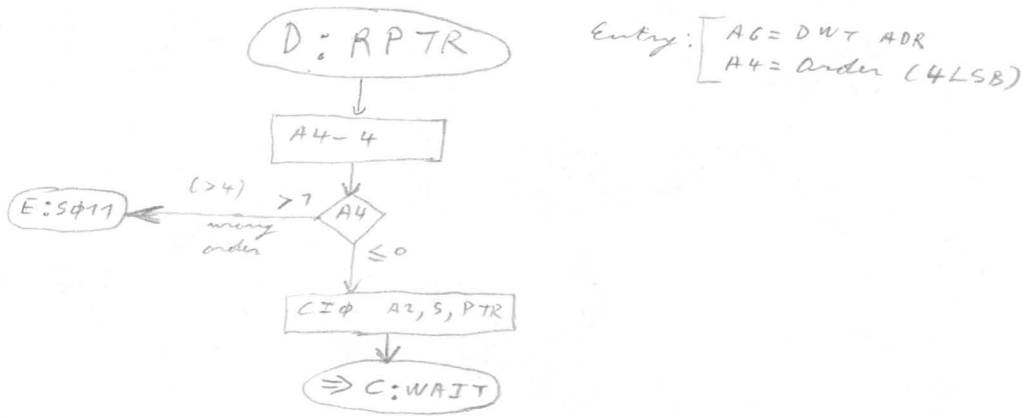
DRAS02:
A2 ⇐ /FFFE
·5⇒ 12, A6 Buff Addr
A2 ⇐ ((12+A6)) Counter clear. (1st word of Buff)
A4?
A2 ⇐ 2
+5⇒ 12, A6 Buff Addr.
+5→ 16, A6 Eff length
A4 ÷ /FE

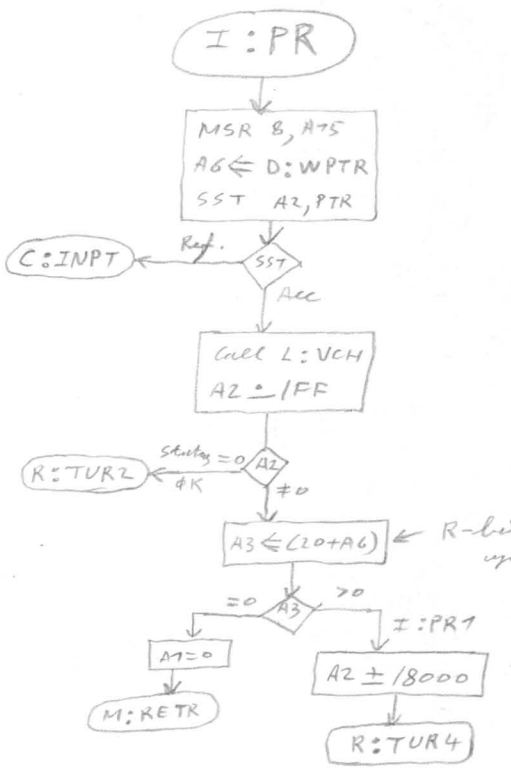
A4 ⇐6
14+A6 ← Req length
A4 ⇐5
18+A6 ← Order

DRAS03:
A4 ≠130
A4 =130
A7 ⇐ /0A
A7 ⇐0

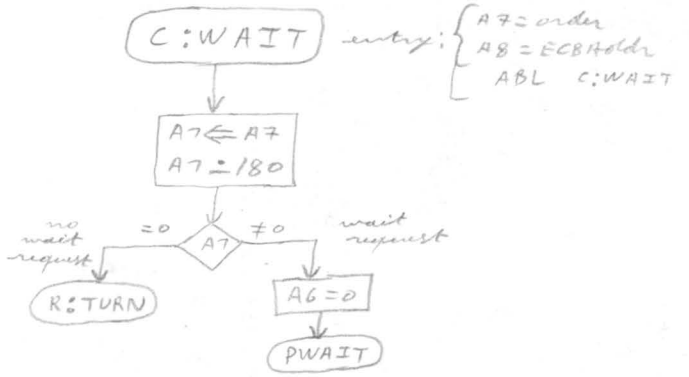
DRAS04:
A2 ⇐0 ← output
INH
CIφ A2, 5, TY
φTR A7, 0, TY
EF A75, φ:TPUT
WAITTS
→ C:WAIT

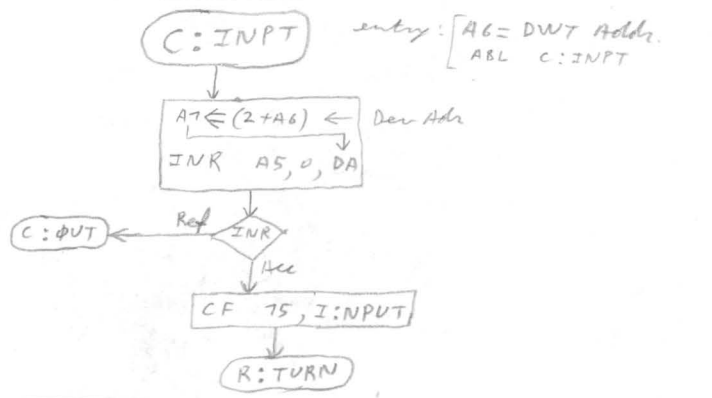
open
- D:RAS2
  ↳ ASR reader
- D:RAS3
  ↳ ASR punch

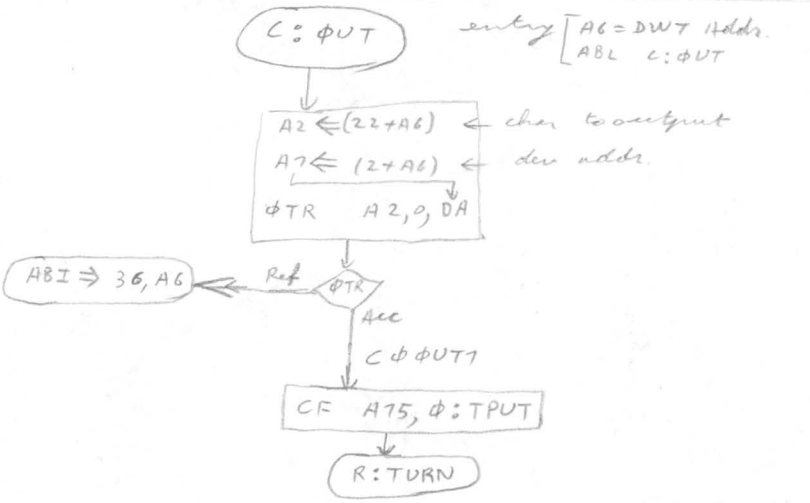

D:RPTR
Entry: AG = DWT ADR
A4 = order (4 LSB)
A4 - 4
A4
>1
(>4)
wrong order
E:SØ11
≤0
CIØ A2,S,PTR
⇒ C:WAIT

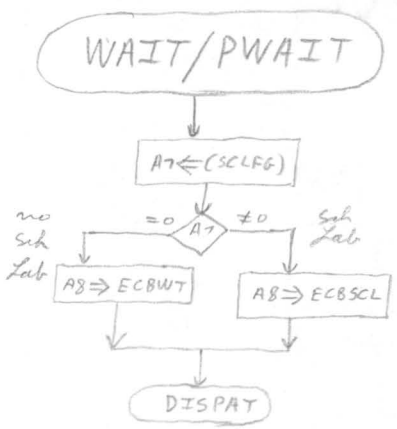

I:PR
MSR 8,A15
AG ⇐ D:WPTR
SST A2,PTR
SST
Ref.
C:INPT
Acc
Call L:VCH
A2 = 1FF
A2
Status = 0
4K
R:TUR2
≠0
A3 ⇐ (20+AG)
R-bit of order in A7 upon LKM call
A3
=0
>0
A1=0
M:RETR
I:PR1
A2 ± 18000
R:TUR4

CΦMIΦ

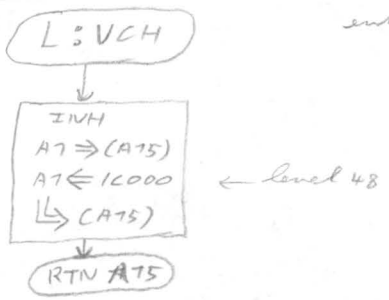

C:WAIT
entry: A7=order
A8=ECBHoldr
ABL C:WAIT
A7⇐A7
A7=180
no wait request
=0
A7
≠0
wait request
R:TURN
A6=0
PWAIT


C:INPT
entry: A6=DWT Addr.
ABL C:INPT
A7⇐(2+A6) ← Dev Adr
INR A5,0,DA
Ref
C:ΦUT
INR
Acc
CF 75,I:NPUT
R:TURN


C:ΦUT
entry A6=DWT Addr.
ABL C:ΦUT
A2⇐(22+A6) ← char to output
A7⇐(2+A6) ← dev addr
ΦTR A2,0,DA
ABI⇒36,A6
Ref
ΦTR
Acc
CΦΦUT1
CF A75,Φ:TPUT
R:TURN

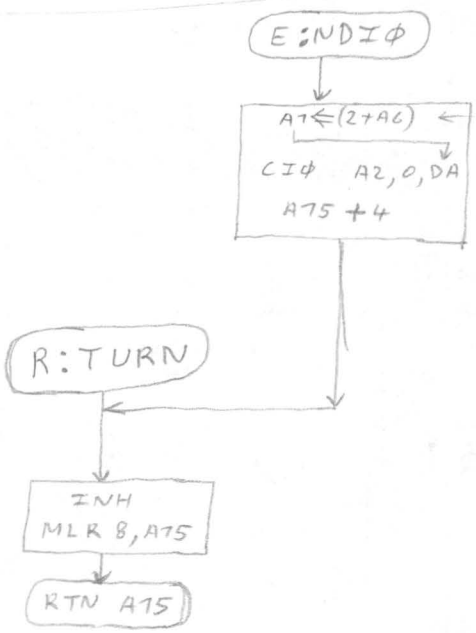


entry: A8 = Controller Status Addr if controller not free
= ECB Addr.
ABL PWAIT



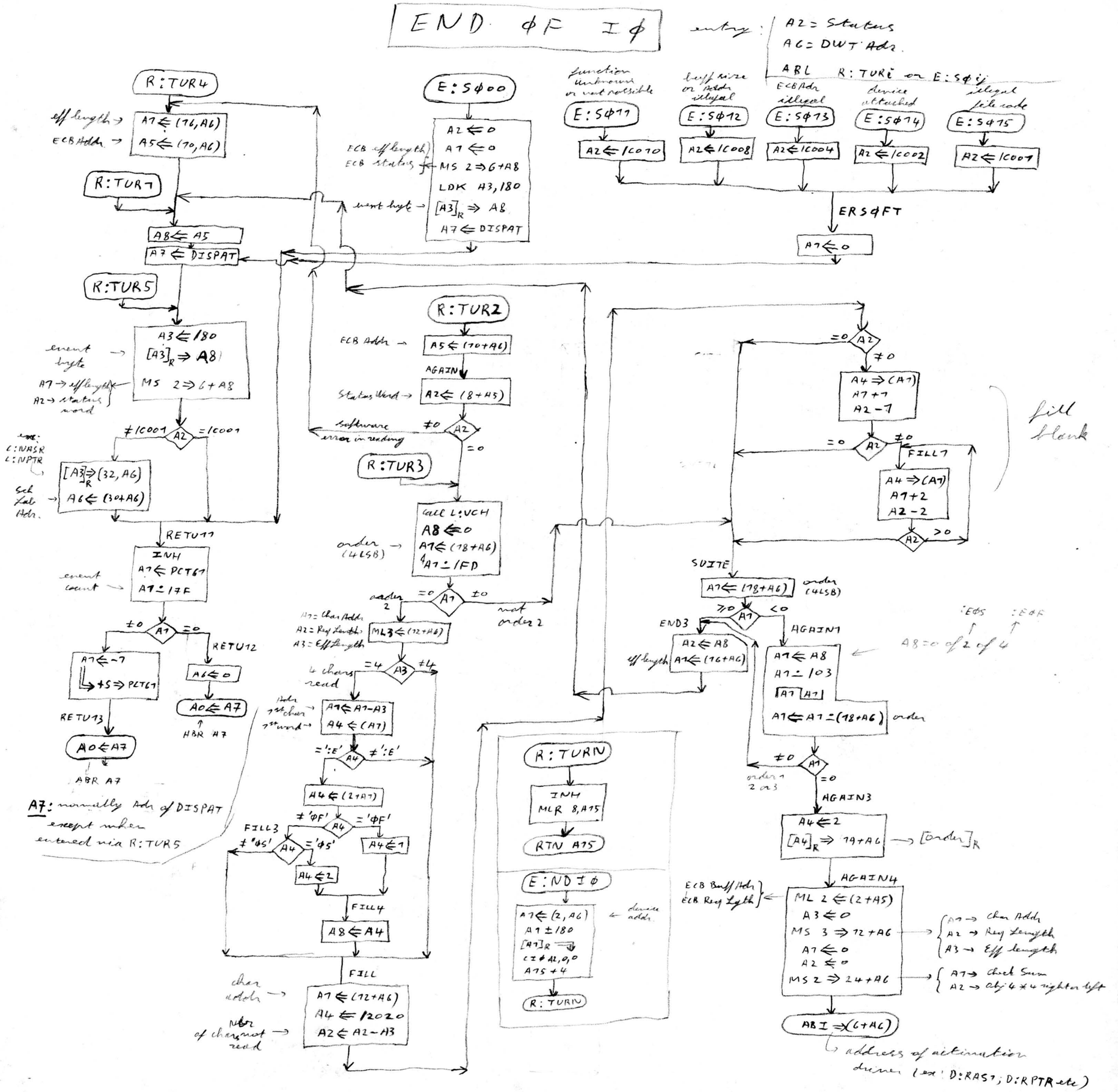
← level 48

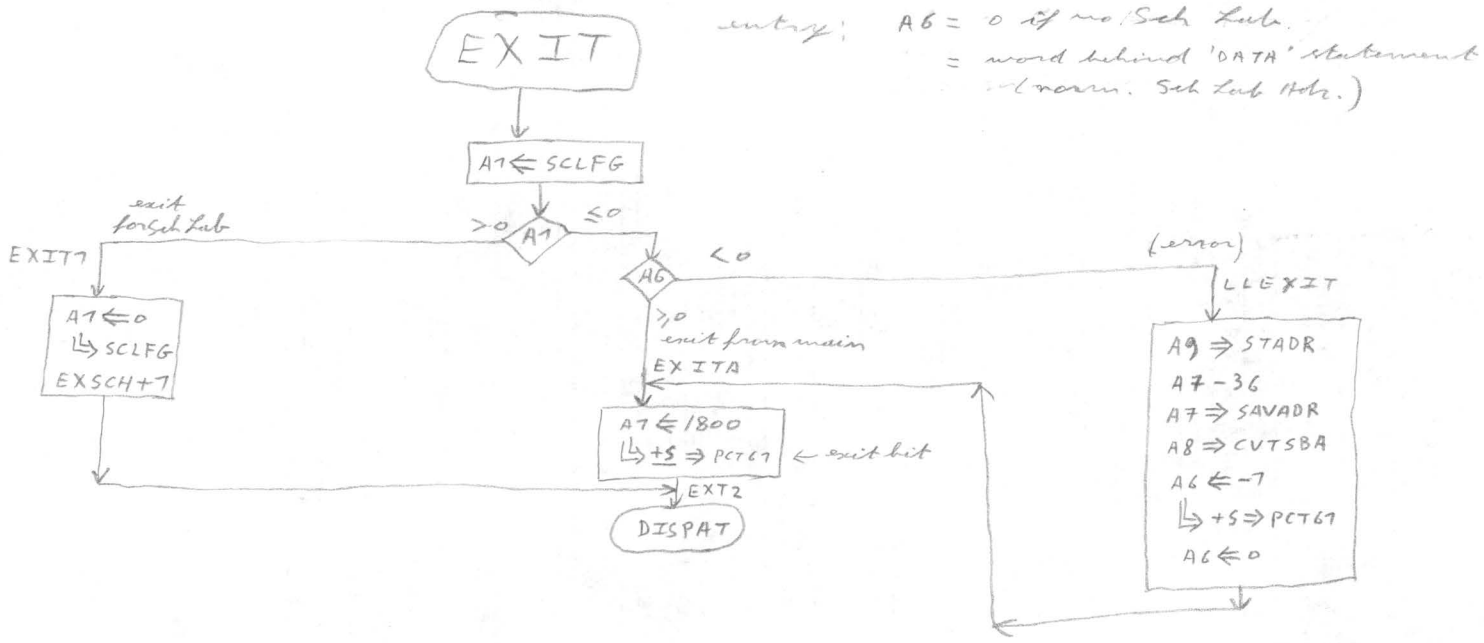
entry: A7 = Return Addr.
ABL L:VCH

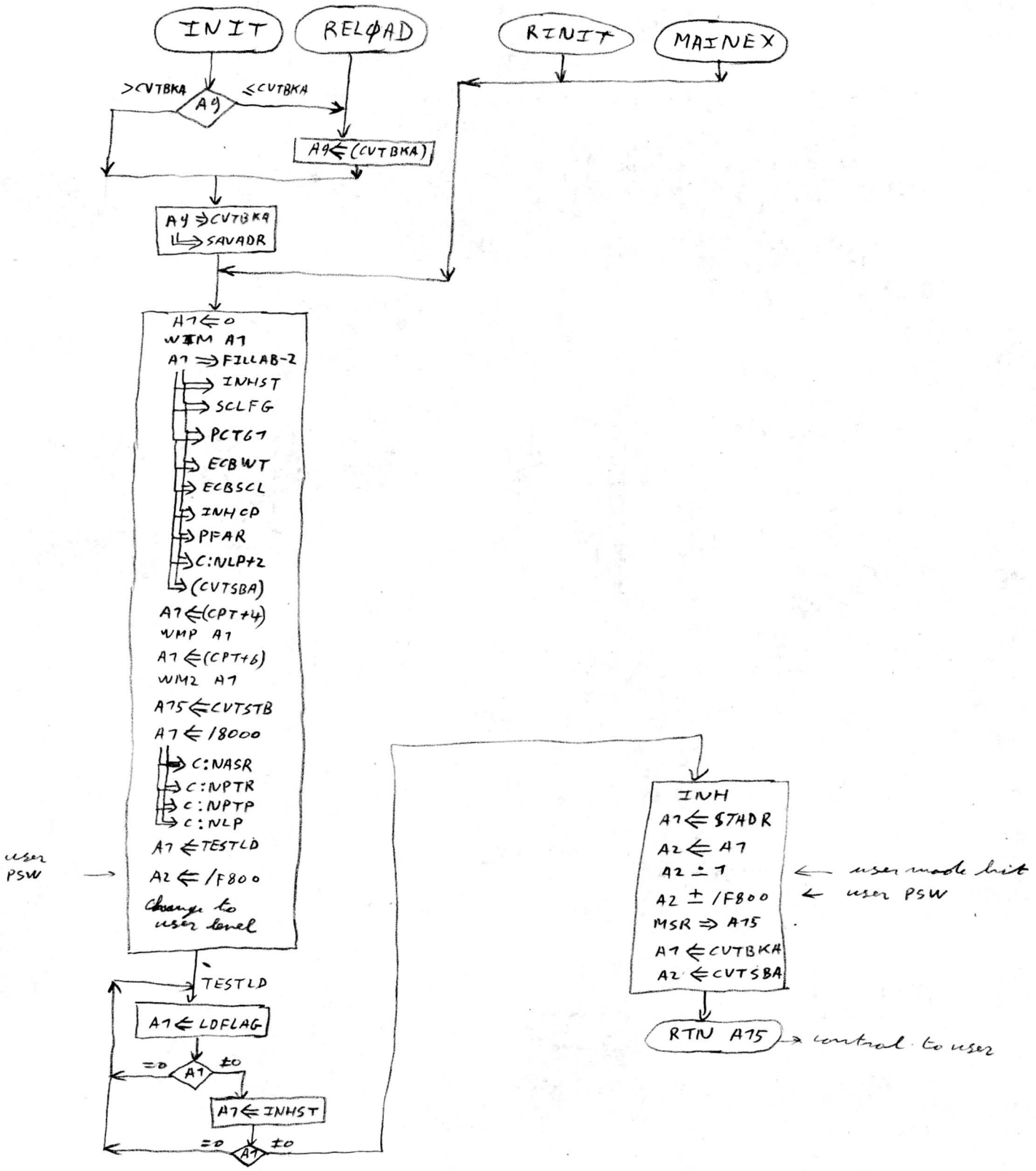


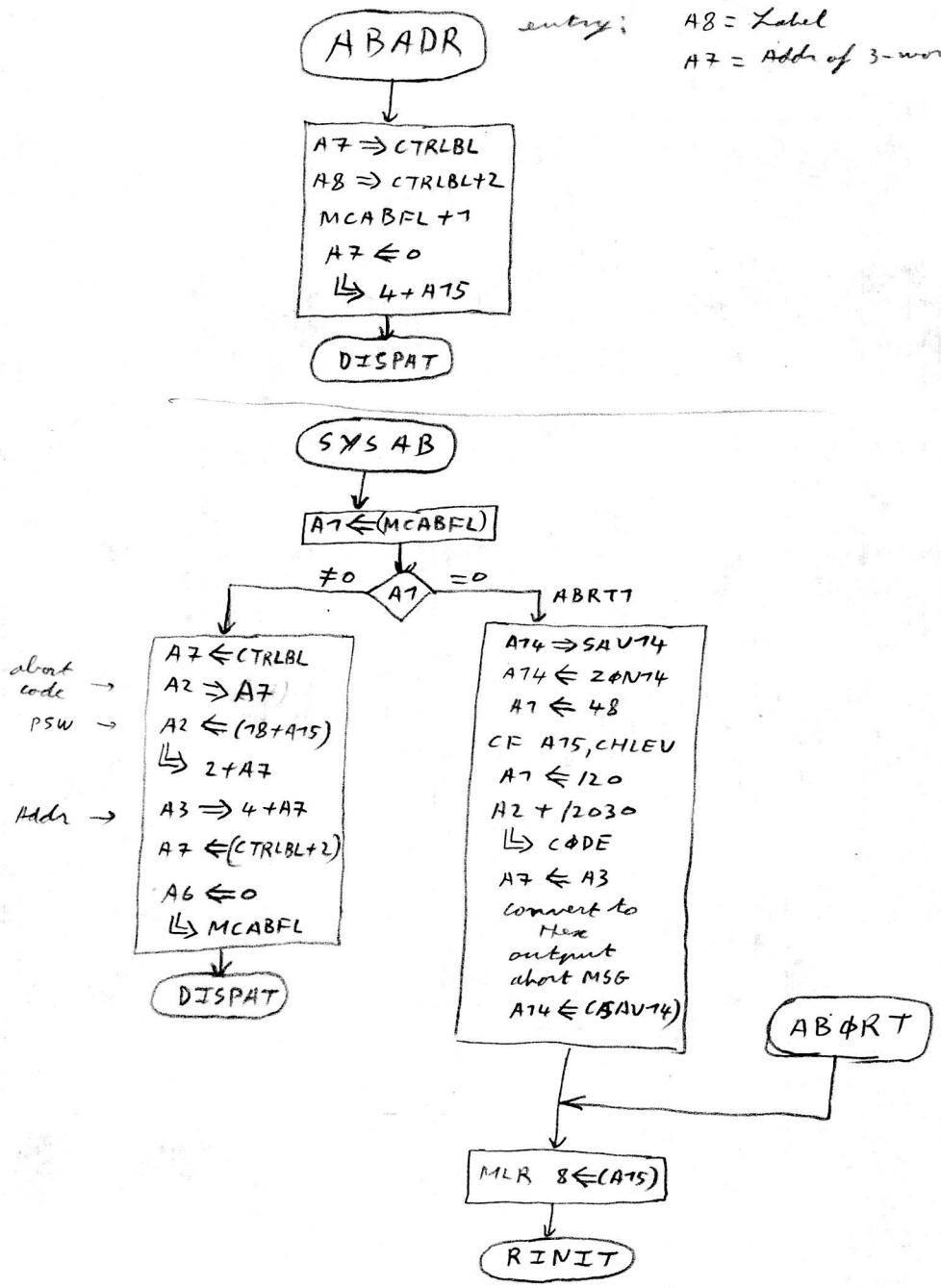
entry AC = DWT Addr
ABL E:NDIΦ

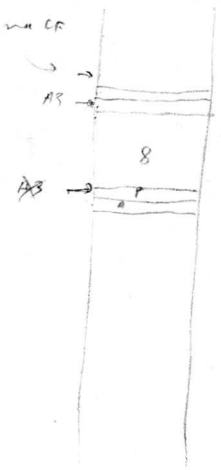
← dev Addr.


END ΦF IΦ
entry: A2 = Status
A6 = DWT Adr.
ABL R:TUR1 or E:SΦ1
R:TUR4
eff length →
ECB Addr →
A7 ⇐ (76,A6)
A5 ⇐ (70,A6)
R:TUR7
AB ⇐ A5
A7 ⇐ DISPAT
R:TUR5
A3 ⇐ 180
event byte →
[A3]R ⇒ A8
A7 → eff length
A2 → status word
MS 2 ⇒ 6+A8
≠ IC001
A2
= IC001
ex: C:IVASR C:INPTR
[A3]R ⇒ (32,A6)
ECB Link Addr →
A6 ⇐ (30+A6)
RETU11
INH
event count →
A7 ⇐ PCT61
A7 ± 17F
±0
A7
=0
RETU12
A7 ⇐ -7
+5 ⇒ PCT61
RETU13
A0 ⇐ A7
ABR A7
A6 ⇐ 0
A0 ⇐ A7
ABR A7
A7: normally Adr of DISPAT
except when
entered via R:TUR5
E:SΦ00
A2 ⇐ 0
A7 ⇐ 0
ECB eff length
ECB status
MS 2 ⇒ 6+A8
LDK A3,180
event byte →
[A3]R ⇒ A8
A7 ⇐ DISPAT
function unknown or not possible
E:SΦ11
A2 ⇐ IC010
buff size or addr illegal
E:SΦ12
A2 ⇐ IC008
ECB Adr illegal
E:SΦ13
A2 ⇐ IC004
device attached
E:SΦ14
A2 ⇐ IC002
illegal file code
E:SΦ15
A2 ⇐ IC007
ERSΦFT
A7 ⇐ 0
R:TUR2
ECB Addr →
A5 ⇐ (70+A6)
AGAIN
Status Word →
A2 ⇐ (8+A5)
software error in reading
≠0
A2
=0
R:TUR3
call L:VCH
A8 ⇐ 0
order (4 LSB) →
A7 ⇐ (78+A6)
ΦA7 ± 1FD
order 2
=0
A7
±0
not order 2
A7 = Char Addr
A2 = Req Length
A3 = Eff length
ML3 ⇐ (72+A6)
4 chars read
=4
A3
≠4
Buffer → 1st char → 1st word →
A7 ⇐ A7-A3
A4 ⇐ (A7)
= ':E'
A4
≠ ':E'
A4 ⇐ (2+A7)
≠ 'ΦF'
A4
= 'ΦF'
FILL3
≠ 'Φ5'
A4
= 'Φ5'
A4 ⇐ 7
A4 ⇐ 2
FILL4
A8 ⇐ A4
FILL
char Adds →
Nbr of chars not read →
A7 ⇐ (72+A6)
A4 ⇐ 12020
A2 ⇐ A2-A3
R:TURN
INH
MLR 8,A75
RTN A75
E:NDIΦ
A7 ⇐ (2,A6)
A7 ± 180
[A7]R ⇒
CI ΦA2,0,0
A75+4
device addr.
R:TURN
=0
A2
≠0
A4 ⇒ (A7)
A7 +7
A2 -7
=0
A2
≠0
FILL7
A4 ⇒ (A7)
A7+2
A2 -2
A2
>0
fill blank
SUITE
A7 ⇐ (78+A6)
order (4 LSB)
≥0
A7
<0
END3
A2 ⇐ A8
eff length
A7 ⇐ (76+A6)
AGAIN1
A7 ⇐ A8
A7 ± 103
A7 A7
A7 ⇐ A7 ± (78+A6)
order
≠0
A7
=0
order 1 or 3
AGAIN3
A4 ⇐ 2
[A4]R ⇒ 79+A6
→ [order]R
AGAIN4
ECB Buff Adr
ECB Req Length
ML 2 ⇐ (2+A5)
A3 ⇐ 0
MS 3 ⇒ 72+A6
A7 → char Addr
A2 → Req length
A3 → Eff length
A7 ⇐ 0
A2 ⇐ 0
MS 2 ⇒ 24+A6
A7 → check Sum
A2 → obj 4 x 4 right shift
ABI ⇒ (6+A6)
address of destination driver (ex: D:RAS7; D:RPTR etc)
A8 = 0 of 2 of 4
:EΦS
:EΦF


EXIT
entry: A6 = 0 if no Sch Lab.
= word behind 'DATA' statement (norm. Sch Lab Hdr.)
A7 ⇐ SCLFG
exit for Sch Lab
>0
A7
≤0
EXIT7
A6
<0
(error)
LLEXIT
A7 ⇐ 0
↳ SCLFG
EXSCH+7
≥0
exit from main
EXITA
A9 ⇒ STADR
A7 - 36
A7 ⇒ SAVADR
A8 ⇒ CVTSBA
A6 ⇐ -7
↳ +S ⇒ PCTG7
A6 ⇐ 0
A7 ⇐ 1800
↳ +S ⇒ PCTG7
← exit bit
EXT2
DISPAT


INIT
RELØAD
RINIT
MAINEX
>CVTBKA
A4
≤CVTBKA
A4≤(CVTBKA)
A4⇒CVTBKA
SAVADR
A7≤0
WIM A7
A7⇒FILLAB-2
INHST
SCLFG
PCTG7
ECBWT
ECBSCL
INHCP
PFAR
C:NLP+2
(CVTSBA)
A7≤(CPT+4)
WMP A7
A7≤(CPT+6)
WM2 A7
A75≤CVTSTB
A7≤/8000
C:NASR
C:NPTR
C:NPTP
C:NLP
A7≤TESTLD
user PSW
A2≤/F800
Change to user level
TESTLD
A7≤LDFLAG
=0
A7
±0
A7≤INHST
=0
A7
±0
INH
A7≤$74DR
A2≤A7
A2≠7
A2±/F800
MSR⇒A75
A7≤CVTBKA
A2≤CVTSBA
← user mode bit
← user PSW
RTN A75
→ control to user


ABADR
entry:
A8 = Label
A7 = Addr of 3-word par. 1st: Abort code
2nd: Aborted PSW
3rd: Aborted Address
A7 ⇒ CTRLBL
A8 ⇒ CTRLBL+2
MCABFL +1
A7 ⇐ 0
↳ 4+A15
DISPAT
SYS AB
A7 ⇐ (MCABFL)
≠0
A7
=0
ABRT1
abort code →
PSW →
Addr →
A7 ⇐ CTRLBL
A2 ⇒ A7
A2 ⇐ (1B+A15)
↳ 2+A7
A3 ⇒ 4+A7
A7 ⇐ (CTRLBL+2)
A6 ⇐ 0
↳ MCABFL
DISPAT
A14 ⇒ SAV14
A14 ⇐ ZERN14
A7 ⇐ 48
CF A15,CHLEV
A7 ⇐ 120
A2 + 12030
↳ CODE
A7 ⇐ A3
convert to Hex output abort MSG
A14 ⇐ (SAV14)
ABORT
MLR 8⇐(A15)
RINIT



CHLEV

entry: { A1 = level (normally 48-49)
CR A15 CHLEV

MS 8 ⇒ SAVE
A3 ⇐ A15
A3+2

CHLEV1

A3+20
A2 ⇐ (A3)
keep level in A2

→ search a PSW with level = A1

<A1 A2 ≥A1

A3-78
A2 ⇐ (A15)

→ Adr of an A0

CHLEV2

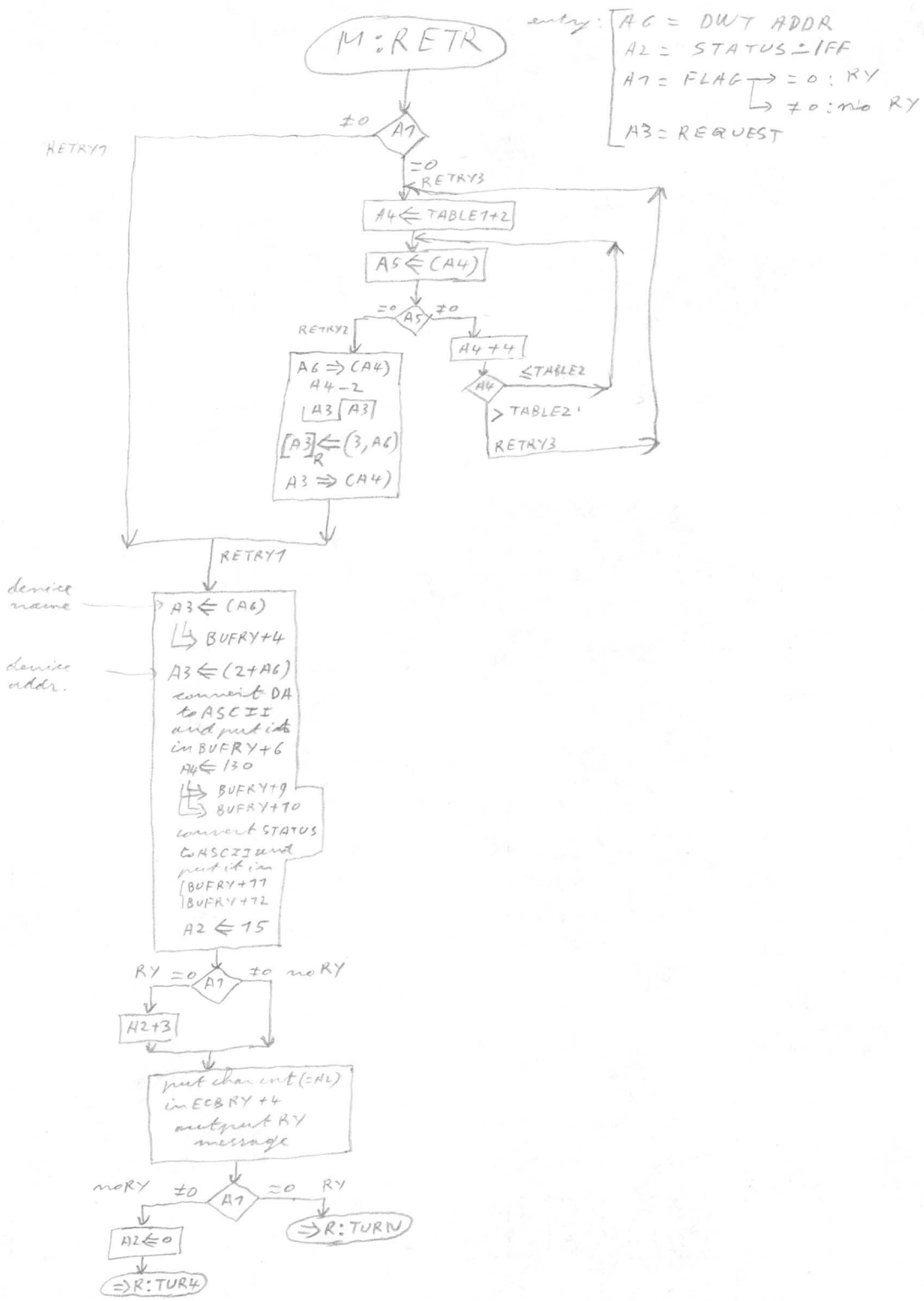
A2+2
A4 ⇐ (A2)
↳ -20+A2

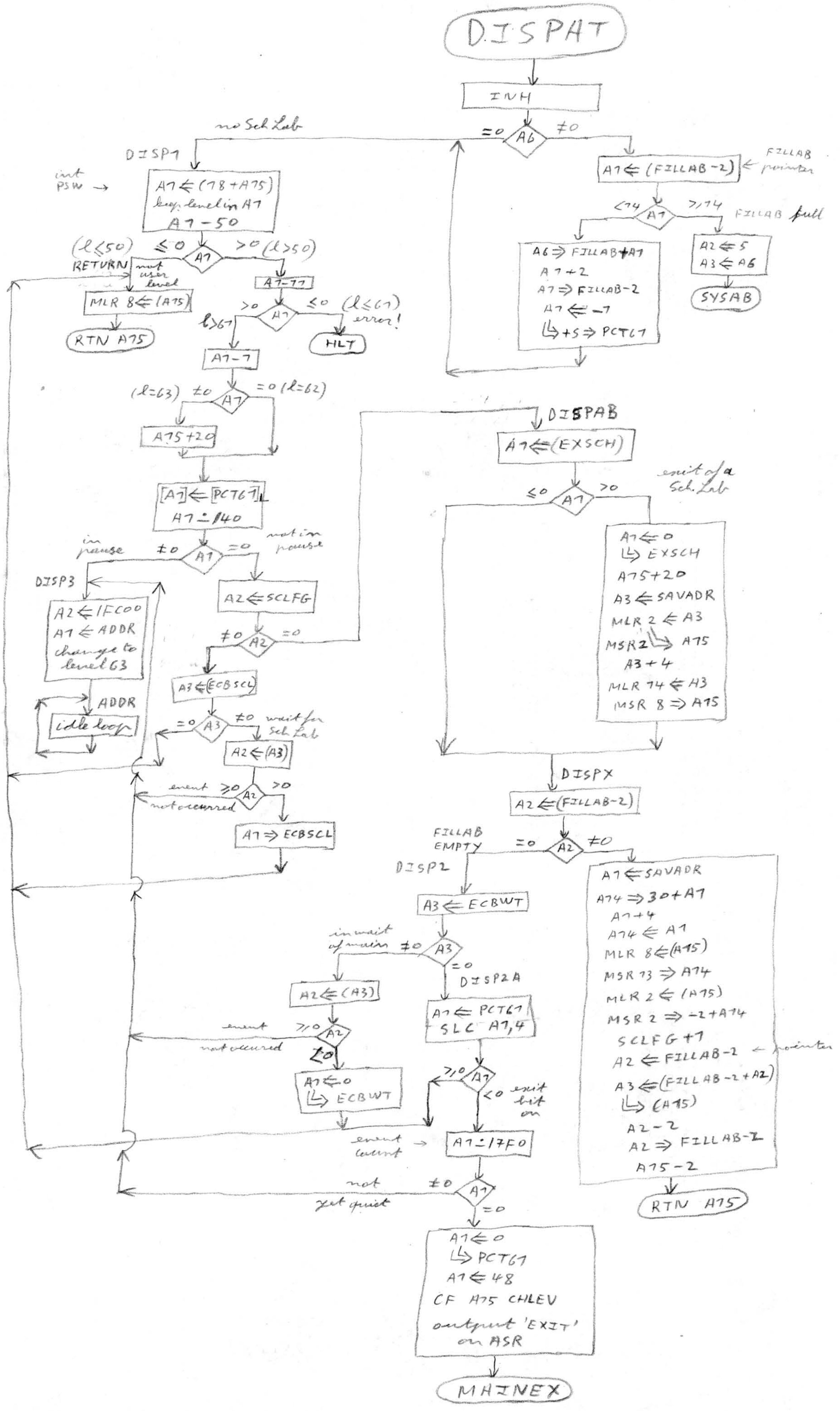
→ move stack point at adr < A3 of 10 words

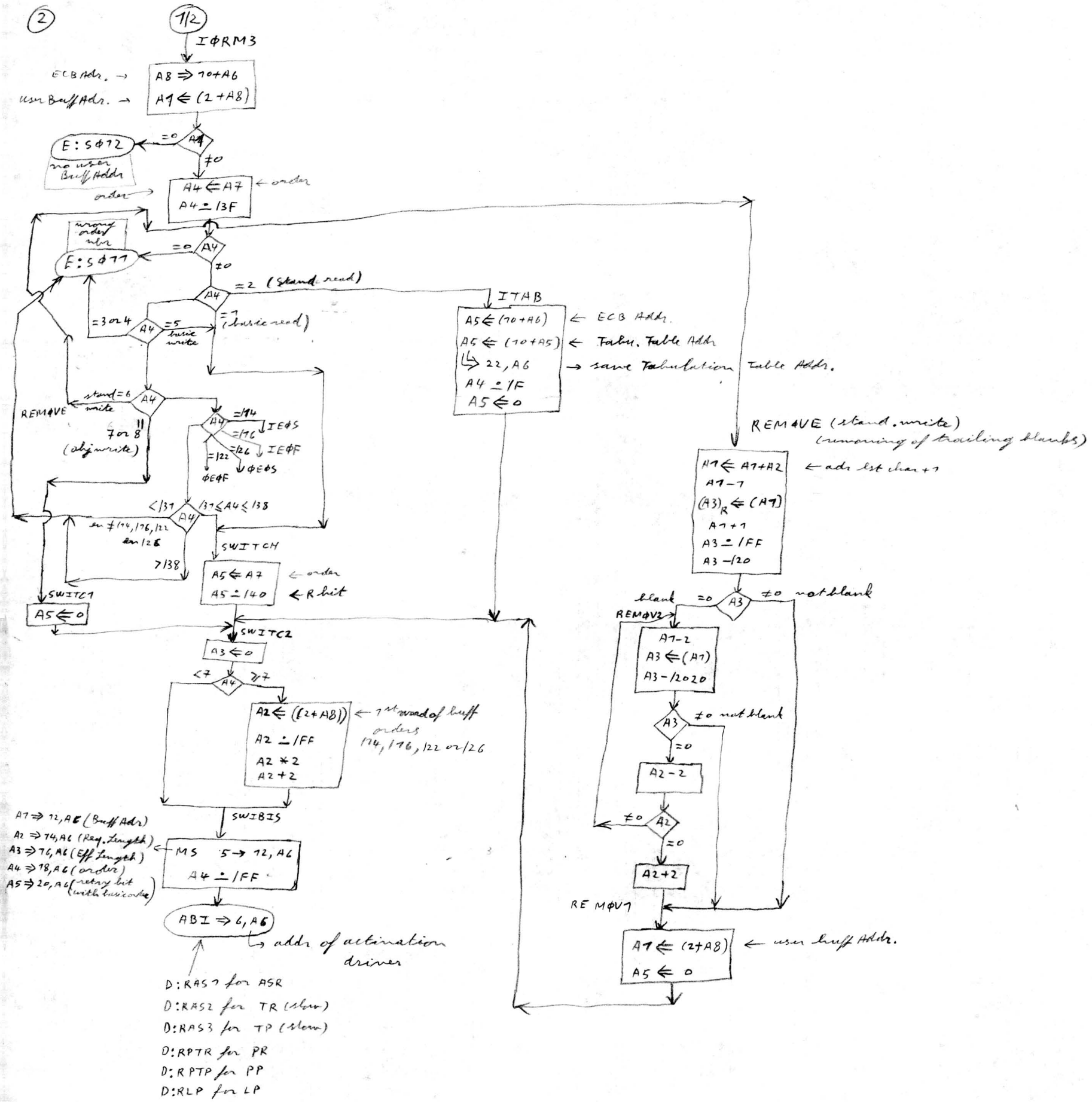
≠A3 A2 =A3

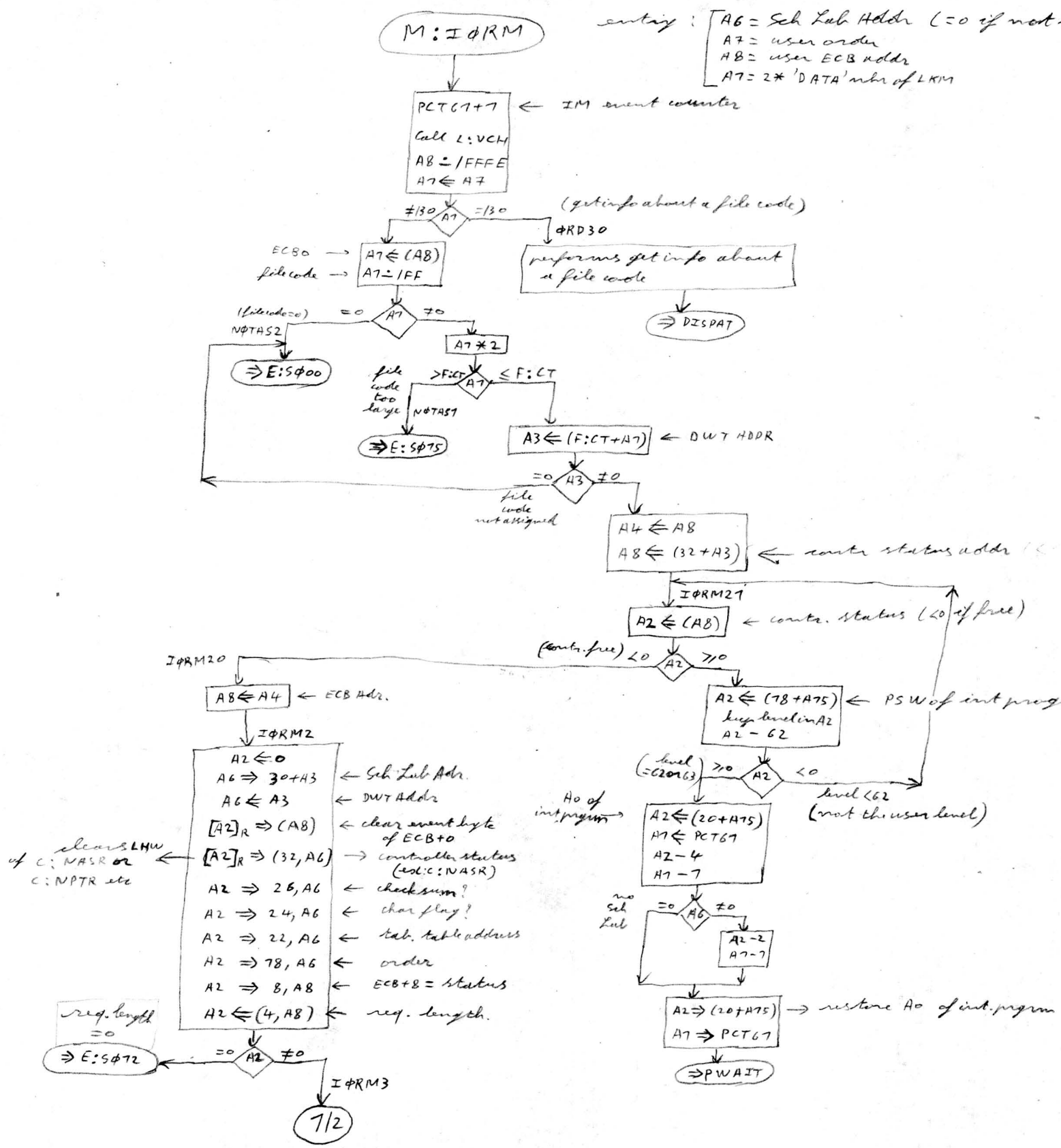
A15-20
A2 ⇐ (4+A15) ← A0 of calling progr.
↳ (A3) ← A0
A2 ⇐ (2+A15) ← PSW of calling progr.
A2 = /3FF
SLL A1,10 ← desired level
A1 ⇐ A1 ± A2
↳ (-2+A3) ← PSW
A15 ⇒ SAV15
A3-4
A15 ⇐ A3
ML 8 ⇐ (SAVE)
MSR 8 ⇒ A15
A15 ⇐ (SAV15)
A15+4
A6 ⇐ 0

DISPAT


M:RETR
entry: A6 = DWT ADDR
A2 = STATUS-IFF
A7 = FLAG → =0 : RY
→ ≠0 : no RY
A3 = REQUEST
≠0
A7
=0
RETRY3
RETRY1
A4 ⇐ TABLE1+2
A5 ⇐ (A4)
=0
A5
≠0
RETRY2
A4+4
A6 ⇒ (A4)
A4-2
A3 A3
[A3]R ⇐ (3, A6)
A3 ⇒ (A4)
A4
≤TABLE2
>TABLE2'
RETRY3
RETRY1
device name
A3 ⇐ (A6)
BUFRY+4
device addr.
A3 ⇐ (2+A6)
convert DA to ASCII and put it in BUFRY+6
A4 ⇐ 130
BUFRY+9
BUFRY+10
convert STATUS to ASCII and put it in
BUFRY+11
BUFRY+12
A2 ⇐ 15
RY =0
A7
≠0 noRY
A2+3
put character count (=A2) in ECB RY+4
output RY message
noRY ≠0
A7
=0 RY
A2 ⇐ 0
⇒R:TUR4
⇒R:TURN


DISPAT
INH
no Sch Lab
=0 A6 ≠0
DISP1
int PSW
A1 ⇐ (18+A15)
A1 − 50
≤0 A1 >0
RETURN
MLR 8 ⇐ (A15)
RTN A15
A1−11
HLT
error!
A1−1
A15+20
[A1] ⇐ [PCT61]
A1 ∸ 140
in pause
not in pause
DISP3
A2 ⇐ IFC00
A1 ⇐ ADDR
change to level 63
ADDR
idle loop
A2 ⇐ SCLFG
A3 ⇐ ECBSCL
wait for Sch Lab
A2 ⇐ (A3)
event not occurred
A1 ⇒ ECBSCL
A1 ⇐ (FILLAB−2)
FILLAB pointer
FILLAB full
A6 ⇒ FILLAB+A1
A1+2
A1 ⇒ FILLAB−2
A1 ⇐ −1
+5 ⇒ PCT61
A2 ⇐ 5
A3 ⇐ A6
SYSAB
DISPAB
A1 ⇐ (EXSCH)
exit of a Sch. Lab
A1 ⇐ 0
EXSCH
A15+20
A3 ⇐ SAVADR
MLR 2 ⇐ A3
MSR 2 ⇒ A15
A3+4
MLR 14 ⇐ A3
MSR 8 ⇒ A15
DISPX
A2 ⇐ (FILLAB−2)
FILLAB EMPTY
DISP2
A3 ⇐ ECBWT
in wait of main
DISP2A
A2 ⇐ (A3)
A1 ⇐ PCT61
SLC A1,4
event not occurred
A1 ⇐ 0
ECBWT
exit bit on
event count
A1 ∸ 17F0
not yet quit
A1 ⇐ 0
PCT61
A1 ⇐ 48
CF A15 CHLEV
output 'EXIT' on ASR
MAINEX
A1 ⇐ SAVADR
A14 ⇒ 30+A1
A1+4
A14 ⇐ A1
MLR 8 ⇐ (A15)
MSR 13 ⇒ A14
MLR 2 ⇐ (A15)
MSR 2 ⇒ −2+A14
SCLFG+1
A2 ⇐ FILLAB−2
pointer
A3 ⇐ (FILLAB−2+A2)
(A15)
A2−2
A2 ⇒ FILLAB−2
A15−2
RTN A15


(2)
1/2
IØRM3
ECB Adr. →
User Buff Adr. →
A8 ⇒ 10+A6
A1 ∈ (2+A8)
A4
=0
≠0
E:SØ72
no user Buff Addr
order →
A4 ∈ A7
A4 ≐ 13F
← order
wrong order nbr
E:SØ77
A4
=0
≠0
=2 (stand. read)
=3 or 4
=5 twice write
= 7 (twice read)
ITAB
A5 ∈ (10+A6)
← ECB Addr.
A6 ∈ (10+A5)
← Tabu. Table Addr
↳ 22, A6
→ save Tabulation Table Addr.
A4 ≐ 1F
A5 ∈ 0
stand=6 write
REMØVE
7 or 8 (obj write)
=174
=176
=122
=126
IEØS
IEØF
ØEØS
ØEØF
<137
137≤A4≤138
≠174, 176, 122 or 126
>138
SWITCH
A5 ∈ A7
A5 ≐ 140
← order
← R bit
SWITC1
A5 ∈ 0
SWITC2
A3 ∈ 0
<7
A4
>7
A2 ∈ ((2+A8))
← 1st word of buff orders 174, 176, 122 or 126
A2 ≐ 1FF
A2 * 2
A2 + 2
SWIBIS
A7 ⇒ 12, A6 (Buff Adr)
A2 ⇒ 14, A6 (Req. Length)
A3 ⇒ 16, A6 (Eff Length)
A4 ⇒ 18, A6 (order)
A5 ⇒ 20, A6 (retry bit with twice order)
MS 5→ 12, A6
A4 ≐ 1FF
ABI ⇒ 6, A6
→ addr of activation driver
D:RAS1 for ASR
D:RAS2 for TR (slow)
D:RAS3 for TP (slow)
D:RPTR for PR
D:RPTP for PP
D:RLP for LP
REMØVE (stand. write)
(removing of trailing blanks)
A1 ∈ A1+A2
← adr lst char+1
A1-1
(A3)R ∈ (A1)
A1+1
A3 ≐ 1FF
A3 -120
blank
=0
A3
≠0 not blank
REMØV2
A1-2
A3 ∈ (A1)
A3 -12020
A3
≠0 not blank
=0
A2-2
≠0
A2
=0
A2+2
REMØV1
A1 ∈ (2+A8)
← user Buff Addr.
A5 ∈ 0


M:IØRM
entry : A6 = Sch Lab Addr (=0 if not requested)
A7 = user order
A8 = user ECB addr
A7 = 2* 'DATA' nbr of LKM
PCT67+7 ← IM event counter
Call L:VCH
AB = /FFFE
A7 ⇐ A7
≠130 A7 =130
(get info about a file code)
ØRD30
ECB0 → A7 ⇐ (A8)
file code → A7 = /FF
performs get info about a file code
⇒ DISPAT
(file code=0) =0 A7 ≠0
NØTAS2
A7 * 2
⇒ E:SØ00
file code too large
>F:CT A7 ≤ F:CT
NØTAS7
⇒ E:SØ75
A3 ⇐ (F:CT+A7) ← DWT ADDR
=0 A3 ≠0
file code not assigned
A4 ⇐ A8
A8 ⇐ (32+A3) ← contr. status addr
IØRM27
A2 ⇐ (A8) ← contr. status (<0 if free)
(contr. free) <0 A2 ≥0
IØRM20
A8 ⇐ A4 ← ECB Adr.
IØRM2
A2 ⇐ 0
A6 ⇒ 30+A3 ← Sch Lab Adr.
A6 ⇐ A3 ← DWT Addr
[A2]R ⇒ (A8) ← clear event byte of ECB+0
[A2]R ⇒ (32, A6) → controller status (ex: C:NASR)
clears LHW of C:NASR or C:NPTR etc
A2 ⇒ 26, A6 ← checksum?
A2 ⇒ 24, A6 ← char flag?
A2 ⇒ 22, A6 ← tab. table address
A2 ⇒ 78, A6 ← order
A2 ⇒ 8, A8 ← ECB+8 = status
A2 ⇐ (4, A8) ← req. length.
req. length =0
⇒ E:SØ72
=0 A2 ≠0
IØRM3
7/2
A2 ⇐ (78+A75) ← PSW of int progr.
high level in A2
A2 - 62
(level =62 or 63) ≥0 A2 <0
level <62 (not the user level)
A0 of int prgrm
A2 ⇐ (20+A75)
A7 ⇐ PCT67
A2 - 4
A7 - 7
no Sch Lab
=0 A6 ≠0
A2 - 2
A7 - 7
A2 ⇒ (20+A75) → restore A0 of int.prgrm
A7 ⇒ PCT67
⇒ PWAIT

A7 = Char Addr
A2 = Req Length
A3 = Eff Length
A4 = order (4 LSB)

I:NPUT

entry: A5 = char wich has been input
A6 = DWT
CF A15, I:NPUT

ML 4 (12+A6)
A4 ∸ 1FF
A4 - 7
A4 * 2

ABI SWFUNI+A4

order: 1 2 3 4

BINIP
[A5]R ⇒ A7
A7+7
A3+7

BINIEN
≠A3 A2 =A3
MS 3 ⇒ 12+A6
RTN A15

ENDINP
MS 3 ⇒ 12+A6
E:NDIφ

ASCINP
A5 ∸ 17F
A4 ⇐ (26+A6)

deleteline code read
≠0 A4 =0
A3-10D
≠0 A5 =0 CR
not CR
A4 ⇐ 0
↳ 26+A6

ASCIN7
≥120 A5 <120
=15E A5 ≠15E
del line
A7 ⇐ A7-A3
A3=0
(26+A6)+7

ASCIEN
=0 A3 ≠0
A7-7
A3-7

ASCIN2
=15F A5 ≠15F
BS

ASCIN3
=17F A5 ≠17F
del
=A3 A2 ≠A3

ASCIN6
=15C A5 ≠15C
tab
fill blanks

no tab
ASCIG6
[A5]R ⇒ A7
A7+7
A3+7

special char

ASCIN1
>0 A3 =0
<178 A5 ≥178
A5-178
A4 ⇐ 1

ASCIN4
≠170 A5 =170
≥174 A5 <174
>4 A5 ≤4
=0

ASCIN8
18+A6+7 → order

ASCIN5
≠10D A5 =10D
not CR

ASCIN9
A5 ∸ 1F
A4 ⇐ 2
18+A6+2 → order

φBJIN4
φBJ INPT 4*4

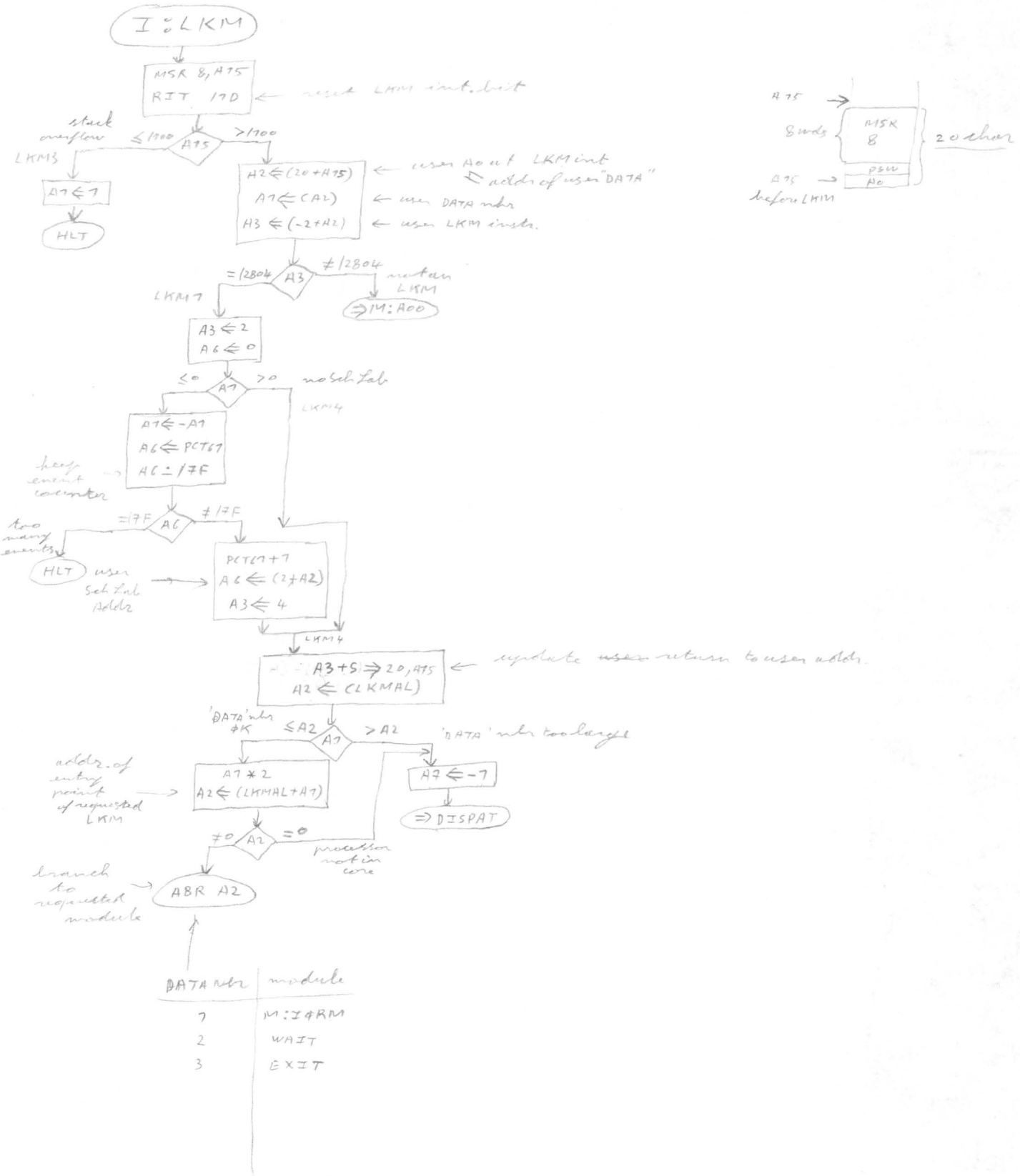
φBJIN8
A4 ⇐ (10+A6) ECB Holder
<U.R.L. A3 >= User Req. Length
[A5]R ⇒ A7
A7+7

φBJM φD
A3+7
A3 =2 φBJI87
≠2

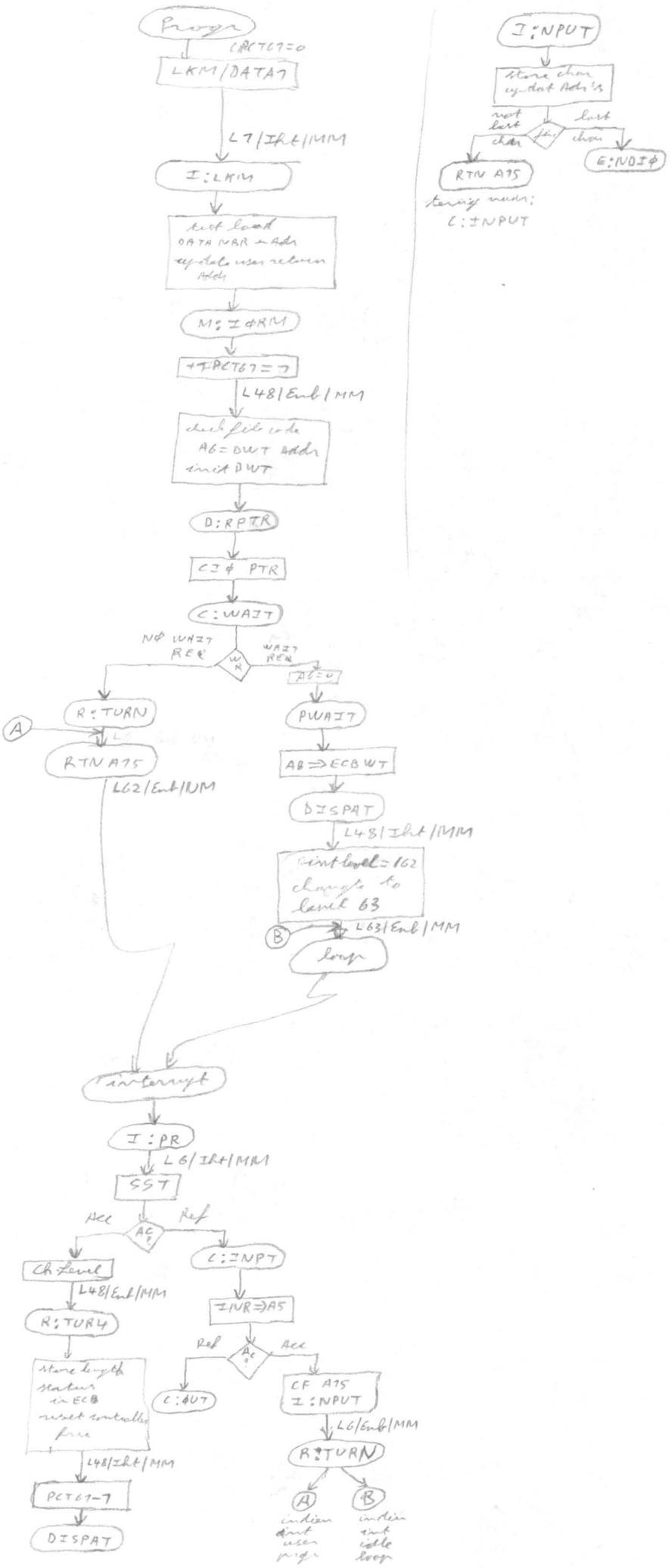
check-sum → A5 ⊕ 5 ⇒ 24+A6
≠A2 A3 =A2
check sum → A5 ⇐ (24+A6)
=0 A5 ≠0
φK
A5 ⇐ 4
↳ +5 ⇒ 8, A4

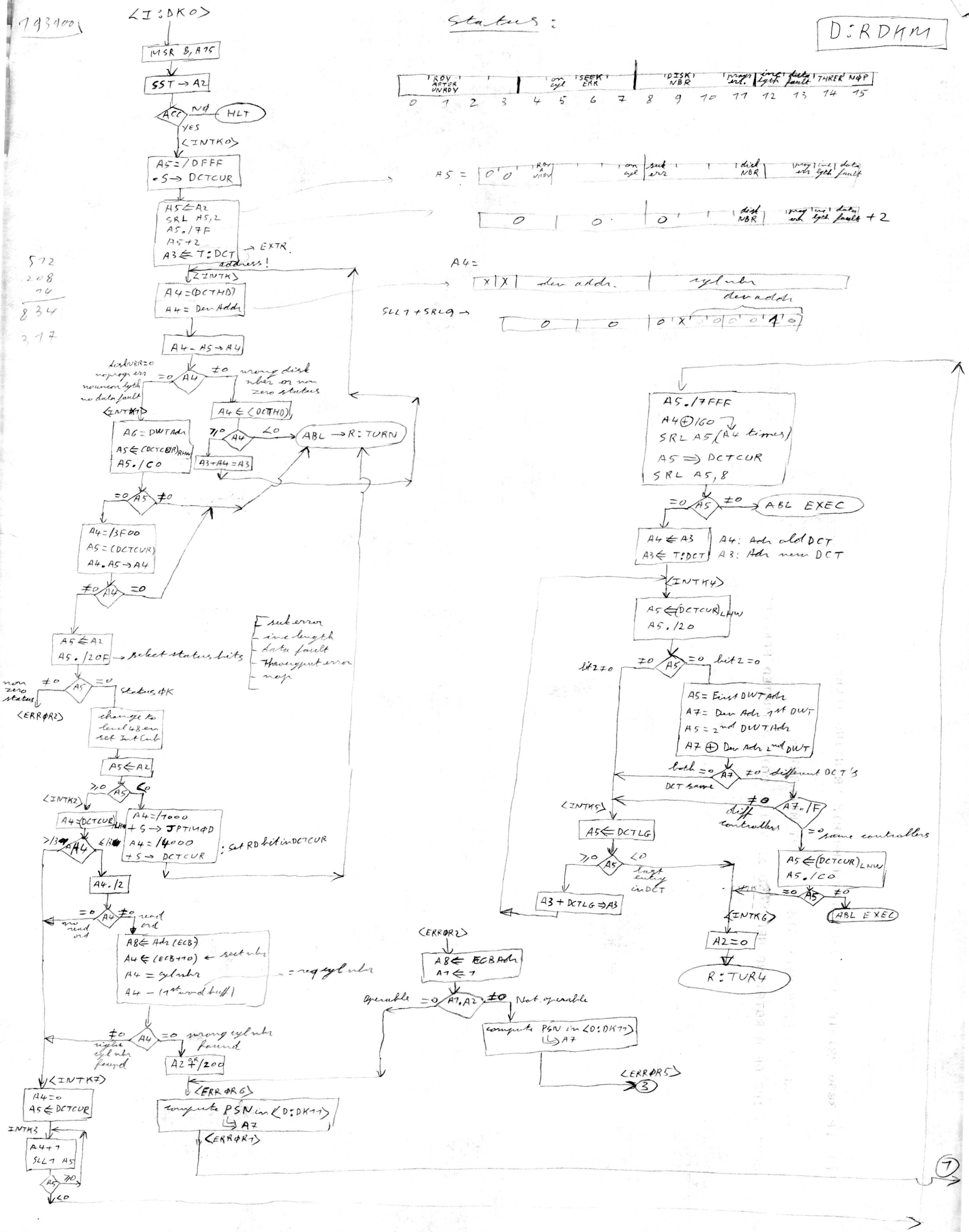
φBJI87
A5 * 2
A5+3
A7 ⇐ A2
A2 ⇐ A5
≤A7 A5 >A7
φBJI83
A5 ⇐ 8
↳ +5 ⇒ 8+A4

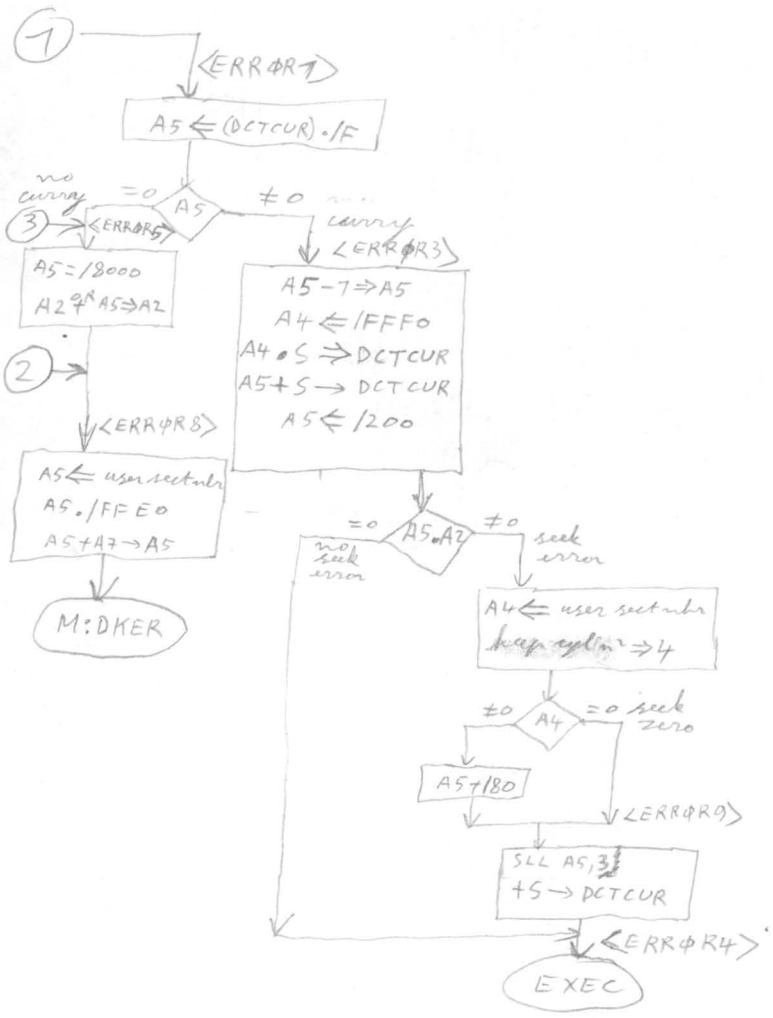
order: 1: basic read
2: ascii read
3: obj (4×4) read
4: obj (8+8) read



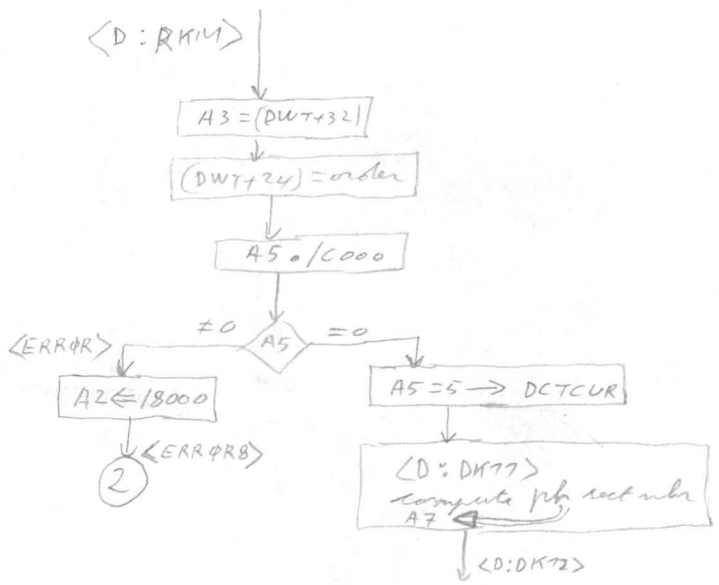
DATA adr	module
1	IM:IARM
2	WAIT
3	EXIT


Progn
LKM/DATA7
L7/IRt/MM
I:LKM
M:IARM
L48/Enb/MM
D:RPTR
C:WAIT
NO WAIT REQ
WAIT REQ
R:TURN
RTN A15
L62/Enb/MM
PWAIT
AB⇒ECB WT
DISPAT
L48/IRt/MM
L63/Enb/MM
loop
interrupt
I:PR
L6/IRt/MM
SST
Acc
Ref
CH Level
L48/Enb/MM
R:TURN
store length
status
in ECB
DISPAT
C:INPT
INR⇒A5
C:OUT
CF A15
I:NPUT
L6/Enb/MM
R:TURN
I:NPUT
store char
RTN A15
C:INPUT
E:NOIP


193400
<I:DKO>
Status:
D:RDKM
MSR 8, A15
SST → A2
ACC
NO
HLT
YES
<INTKO>
RDY AFTER UNRDY
on cyl
SEEK ERR
DISK NBR
prog err.
inc lgth
data fault
THRER
NOP
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
A5=/0FFF
•S→ DCTCUR
A5 =
A5⇐A2
SRL A5,2
A5./7F
A5+2
A3⇐T:DCT
EXTR. address!
+2
A4=
X|X| dev addr. | cyl nbr
dev addr
SLL 1 + SRL 9 →
572
208
74
834
317
<INTK>
A4=(DCTHD)
A4= Dev Addr
A4-A5→A4
disk NBR=0 no prog err no incon lgth no data fault
≠0 wrong disk nbr or non zero status
<INTK1>
A6=DWT Adr
A5⇐(DCTCUR)RHW
A5./C0
A4⇐(DCTHD)
>/0
<0
ABL → R:TURN
A3+A4⇒A3
A4=/3F00
A5=(DCTCUR)
A4.A5→A4
A5.|7FFF
A4⊕/160
SRL A5 (A4 times)
A5 ⇒ DCTCUR
SRL A5,8
=0
≠0
ABL EXEC
A4⇐A3
A3⇐T:DCT
A4: Adr old DCT
A3: Adr new DCT
<INTK4>
A5⇐(DCTCUR)LHW
A5./20
bit2≠0
bit2=0
A5=First DWT Adr
A7= Dev Adr 1st DWT
A5=2nd DWT Adr
A7 ⊕ Dev Adr 2nd DWT
both =0
≠0 different DCT's
DCT same
<INTK5>
A7./F
diff controllers
=0 same controllers
A5⇐DCTLG
last entry in DCT
A3+DCTLG⇒A3
A5⇐(DCTCUR)LHW
A5./C0
ABL EXEC
<INTK6>
A2=0
R:TURN
A5⇐A2
A5./20F → select status bits
seek error
inc length
data fault
Throughput error
nop
non zero status
Status OK
<ERROR2>
change to level 48 en set Int Cub
A5⇐A2
<INTK2>
A4⇐(DCTCUR)LHW
A4=/1000
+S→ JPTIMOD
A4=/4000
+S→ DCTCUR
: set RD bit in DCTCUR
A4./2
=0 no read ord
≠0 read ord
A8⇐Adr (ECB)
A4⇐(ECB+10) ← sect nbr
A4=cyl nbr
A4-(1st word buff)
= req cyl nbr
<ERROR2>
A8⇐ECB Adr
A1⇐1
Operable
A1.A2
Not operable
compute PSN in <D:DK11> ↳ A7
<ERROR5>
3
≠0 right cyl nbr found
=0 wrong cyl nbr found
A2 + or /200
<ERROR6>
compute PSN in <D:DK11> ↳ A7
<ERROR7>
<INTK7>
A4=0
A5⇐DCTCUR
INTK3
A4+1
SLL 1 A5
<0
7


1
<ERR#R1>
A5 ⇐ (DCTCUR).#F
no carry
=0
A5
≠0
carry
3
<ERR#R7>
A5=#8000
A2 OR A5⇒A2
2
<ERR#R3>
A5-1⇒A5
A4⇐#FFF0
A4.S ⇒DCTCUR
A5+S→DCTCUR
A5⇐#200
<ERR#R8>
A5⇐ user sect nbr
A5.#FFE0
A5+A7→A5
M:DKER
=0
A5.A2
≠0
no seek error
seek error
A4⇐ user sect nbr
⇒4
≠0
A4
=0 seek zero
A5+#80
<ERR#R9>
SLL A5,3
+S→DCTCUR
<ERR#R4>
EXEC

D:RDKM



log sect nbr	phy sect nbr
16 0	0
1	3
2	6
3	9
4	12
5	15
6	2
7	
8	
9	
10	
11	
12	
13	
14	
31 15	

head

$PSN = (LSN \times 3)_{mod\,16}$ for a track

{ if track 0 : add 100
{ if track 1 : add 110

<D:DK12> ; entry : A7 = phy sect Nbr (PSN)

A3 = Adr DCTHD

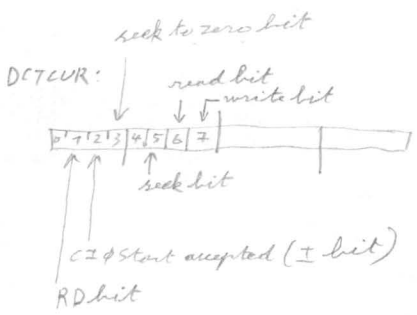
A8 = Adr ECB

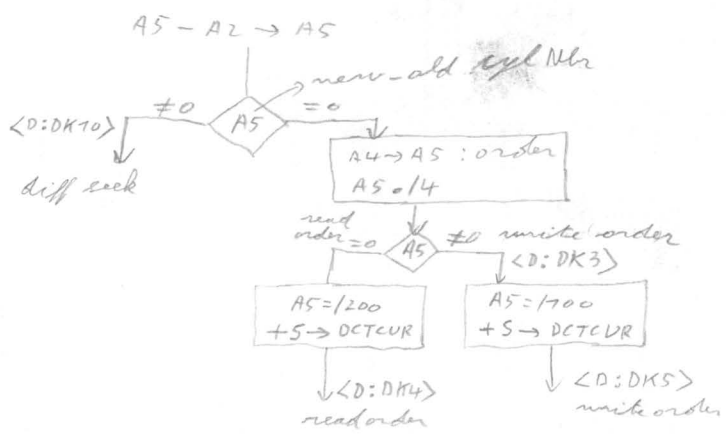
ECB SecN → A5

keep cyl n° → A5

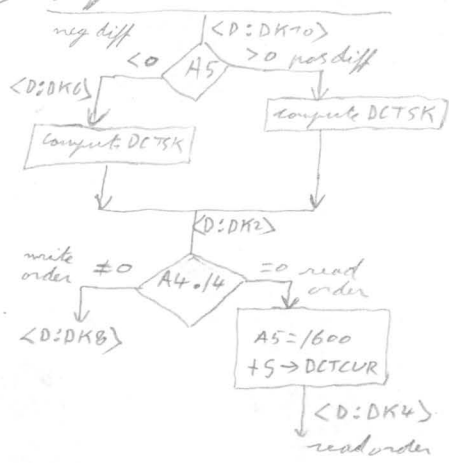
current cyl n° → A2

A5 → $(DCTHD)_{HRW}$; requested sect nbr in $(DCTHD)_{RHW}$

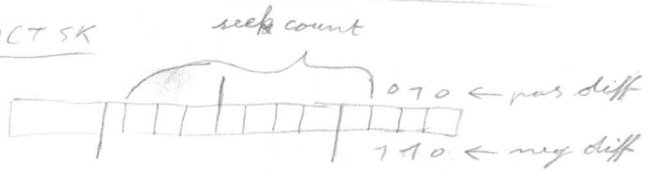




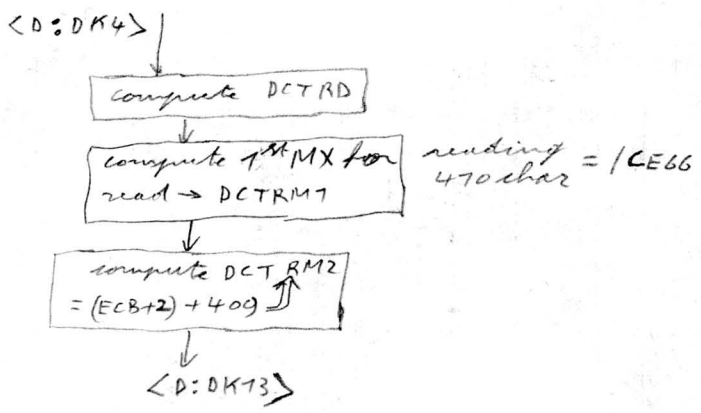
<D:DK10> Differential Seek entry : A5 : cyl diff



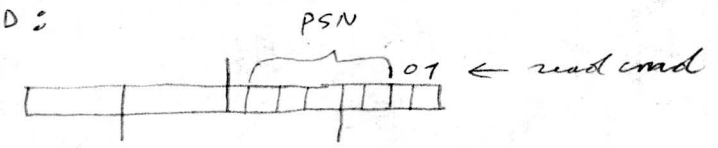
DCTSK



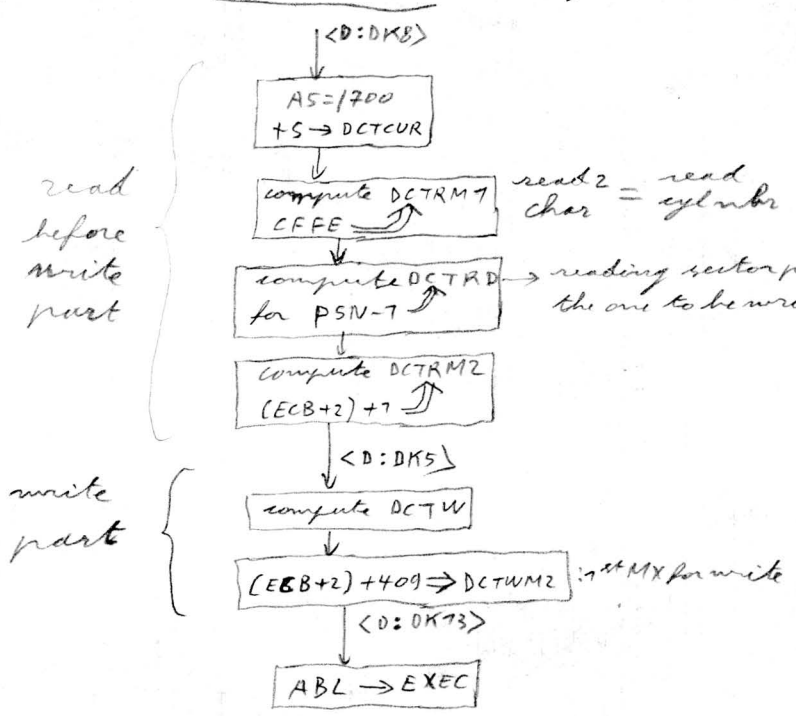
<D:DK4> Read Order entry: A7 = PSN



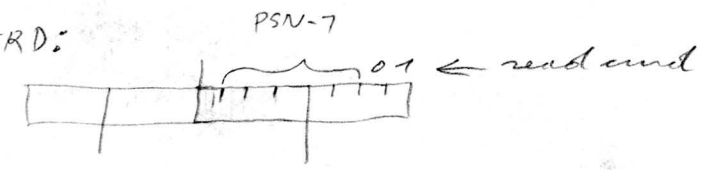
DCTRD:



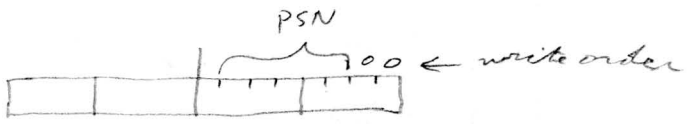
<D:DKB> Write Order entry A7 = PSN

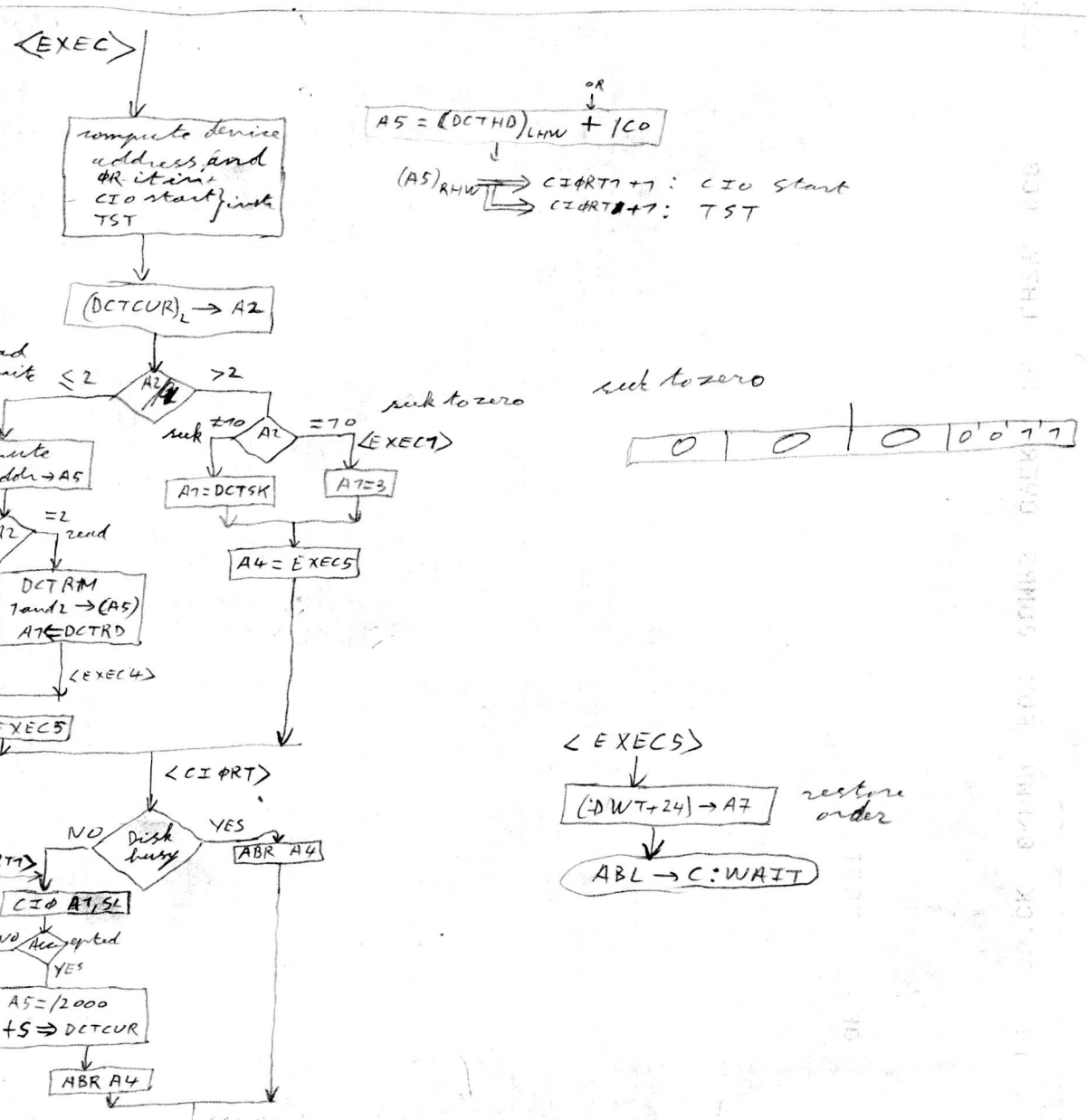


DCTRD:



DCTW





seek to zero