

software X ÷ etc

M:A00 → /24A = 293
 MPYMFD → /007E = 63 356
 DIVMFD → /00AA = 85 447 293
 ADDMFD → /004A = 37 478
 DSUMFD → /0056 = 43 527 words

572
 64
 70
 586

70
 42
 74
 126

760
 70
 790

64
 70
 74

780
 36
 276

M:A00 1

70
 252 ch.
 36 ← 7 blocks
 276

(I: LKM)

MSR 8 → A75
 reset LKM int.
 check stack overflow
 A2 ← user 'DATA' instr
 A7 ← user 'DATA' instr
 A3 ← 'LKM' instr

'LKM' A3 ≠ 'LKM'

<M:A00>
 A4 ← L:TΦPC (= 3)
 A5 ← A3
 (A5)_{RHW} ← (A5)_{LHW}

/DB an /DB en /DB and /DB ≠ A5 = /C00 /C80 /D00 /D8

A3 ← A2
 A2 = 2
 SYSAB

<M:A02>
 A5 ← L:T SVR (= 270)

<M:A03>
 A8 ← T:SVR, A5

load for a free block in T:SVR
 = 0 A8 ≠ 0

<M:A04>
 A8 ← A5
 A8 + Adr(T:SVR) + 4
 MSR 8 → Z:SVR
 MLR 8 ← A75
 MSR 14 → T:SVR + 4 + 78i

A8 = 4
 T:SVR - 4 → T:SVR

user PSW and A0
 monitor reg
 MLR 2 ← A75
 MSR 2 → T:SVR + 0 + 78i
 A7 ← Adr(M: B00)
 A2 ← /FFFF
 MSR 2 → A75
 ML 8 ← Z:SVR
 RTN A75

<M:B00>
 A6 ← A3
 ← /00DF

A6 = 0: constant instr
 ≠ 0 <M:B01>

A6 ← A3
 ← /0002

reg 8-75 ≠ 0 A6 = 0
 A6 ← A3
 A6 ÷ 4
 A6 ÷ 7
 A6 + 8
 A6 X 2

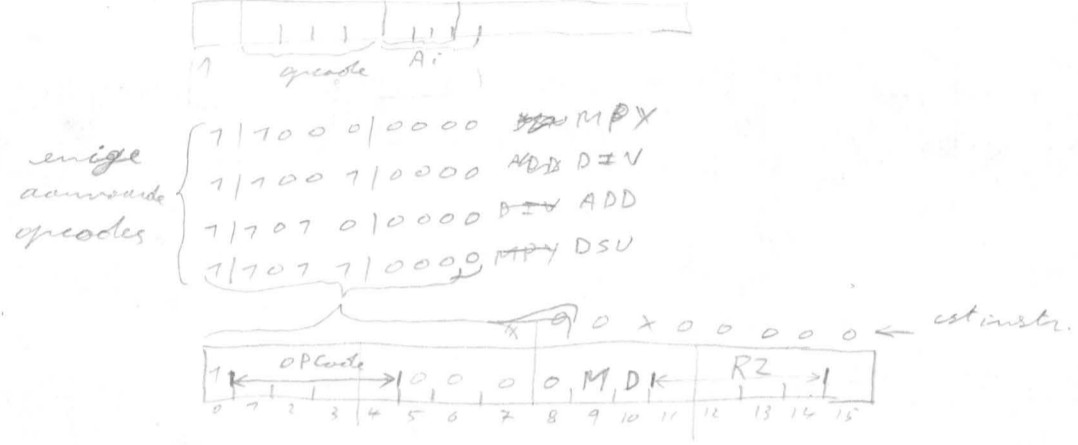
<M:B07A>
 A6 ← A3
 A6 X 2
 A6 ÷ F

A4 = 0007
 MUR DVK <M:B00B>
 A9 = 2
 T:SVR, A5 + 2

A4 = 203
 DAK DSK <M:B00A>
 A9 = 4
 T:SVR, A5 + 4

<M:B06>
 1/2

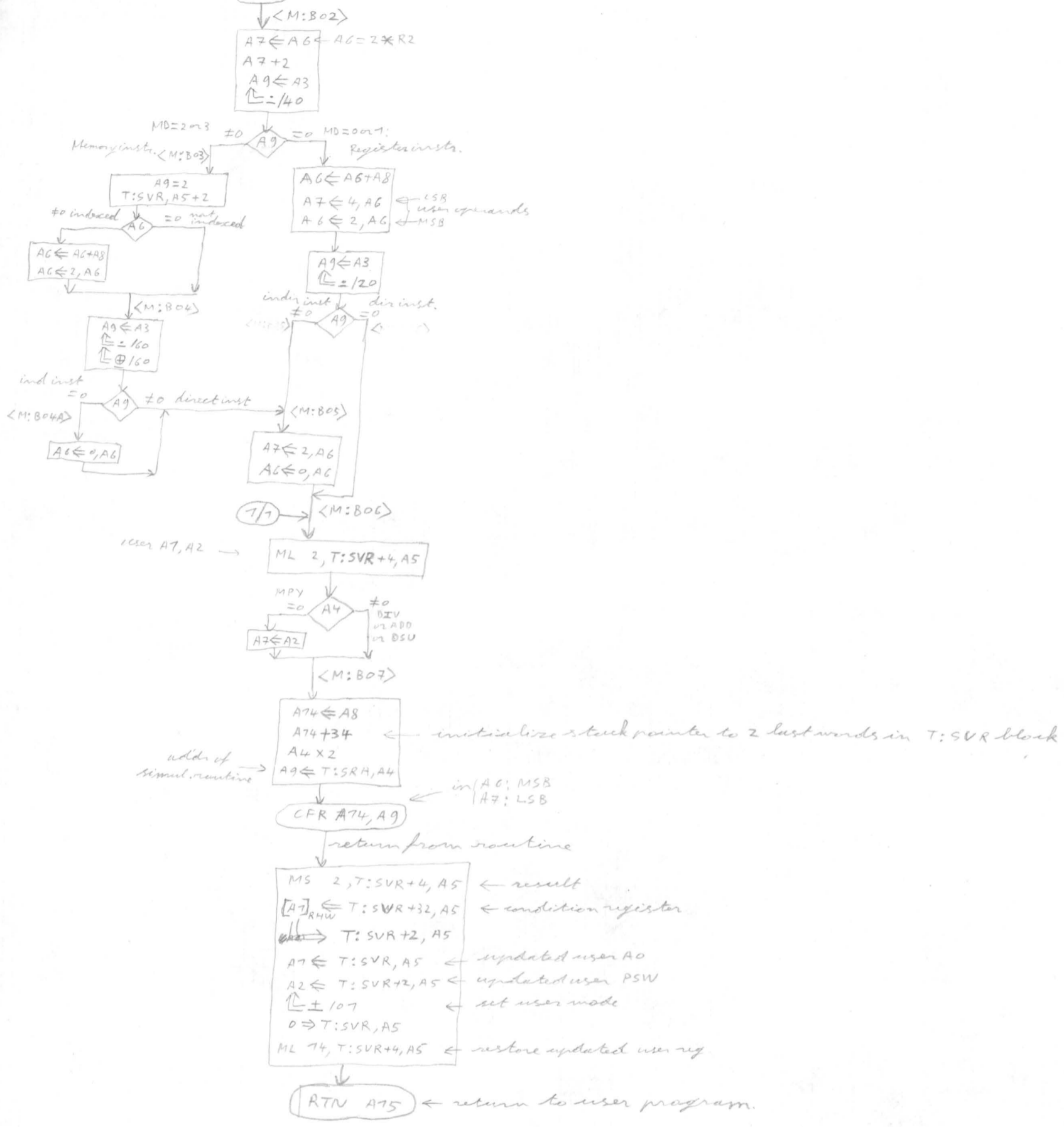
2/2



here A4 = 0: MPY
 1: DIV
 2: ADD
 3: DSU



2/7

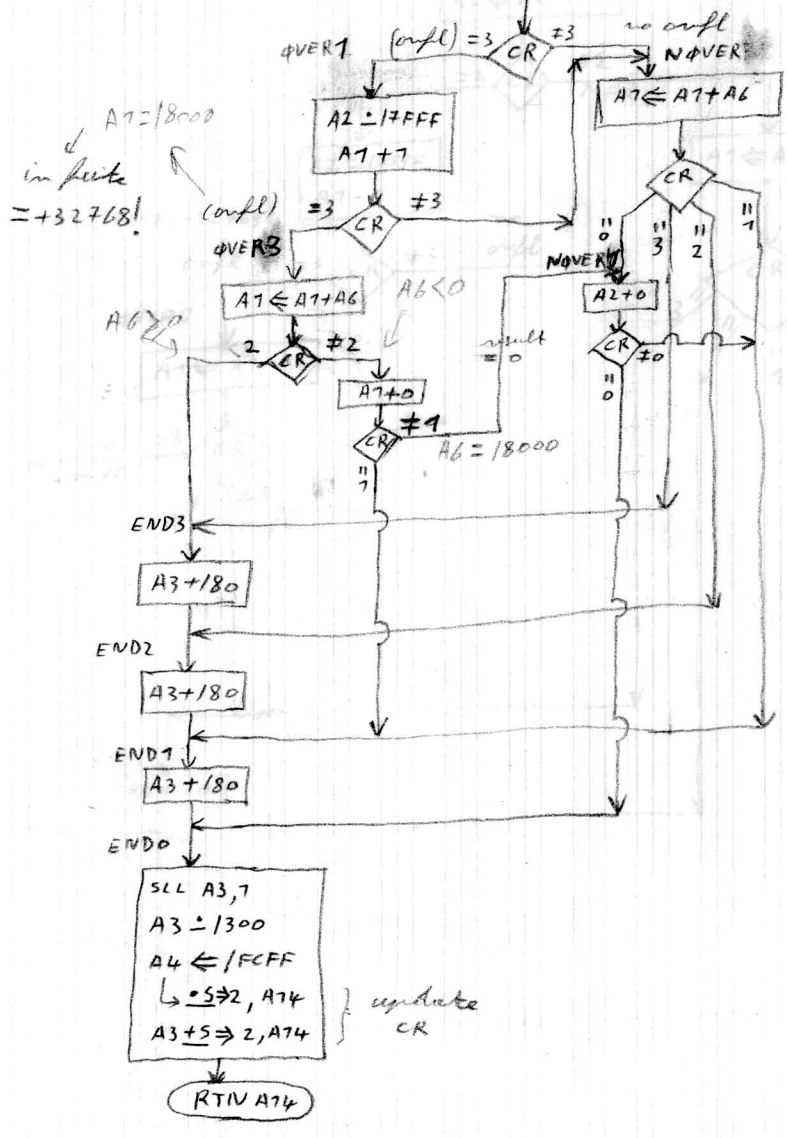


ADD MCD

entry: A7-A2 1st arg
 A6-A7 2nd arg
 CF A74, ADD MCD
 return: A7-A2 result

ADD MCD

A3 ← 0
 A7 ← 17FFF
 A2 ← 17FFF
 A2 ← A2 + A7



*A7 = 18000
 in fact
 = +32768!*

OVER3: A7 = 18000 = +32768!

| A7 + A6 | resulting CR | Result in A7 | Cond. | effective CR |
|--|--------------|--------------|-------------|--------------|
| A6 = 0 | 2 | 18000 | ovfl | 3 ⇒ END3 |
| A6 > 0 | 2 | bit0 = 1 | ovfl | 3 ⇒ END3 |
| A6 < 0 > 18000 " -32768 " = 18000 | 3 | bit0 = 0 | result = A7 | 1 ⇒ END1 |
| | 3 | 10000 | result = 0 | 0 ⇒ NOVER1 |

NOVER1:
 if A7 + A6 = 0 ⇒ result > 0 or = 0 depending on value of A2

| A2 | result |
|-----|--------|
| = 0 | = 0 |
| ≠ 0 | > 0 |

