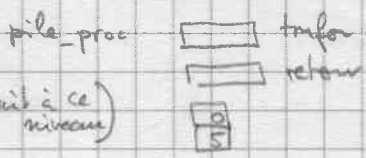


3) Entrée AS pointe au B_debug ou B_init

121
après PB94

exécute un programme basic inséré dans le code 68000

Nota : l'appel doit être fait avec SP=TPPI (car le retour de WSPBAS se fait à ce niveau)



```

WSPBAS1: LEA PB960, AS
WSPBAS = BSR X9R
WSPBAS: BSR DECAN
TST D0
BMI PB96
CMP.B #50, (A3)+
BNE PB96
MEVEN A3
BSR APROC1
MOVE.L IS, (A0)+
MOVE.L (SP)+, (A0)+
MOVER #5, D0
MOVE.L #50000, (A0)+
BRA L250 IRUN
    
```

```

LEA TDEBAS, A0
MOVE.L (A0)+, (A0)+
MOVE.L TPPI, (A0)+
MOVE.L TVALAG, A6
ST YTRACE
BSR PB962
    
```

A0 = pile précédent
tmfor
pc [retour]

→ suite avec appel de proc

```

PB96: RTS
PB960: D.B "B_INIT", 0
PB961: D.B "B_DEBUG", 0
PB962: BSR X9
PB963: BSR APROC2
    
```

```

TDEBAS: D.L 0
D.L 0
D.L 0
    
```

(SP) PB962: LEA TDEBAS, A0
 MOVE.L (A0)+, (A0)+
 MOVE.L TPPI, (A0)+
 MOVE.L TVALAG, A6
 ST YTRACE
 ST BINSTR
 RTS

Conserve TDEBAS et TPPI pour debug
 inhibe bkpt
 inhibe b_instr

3

Sauvegarde avant appel code basic du code 68000

SP

```

PB962: LEA YTRACE, A0
      MOVE.L (A0)+, (A0)
      MOVEQ #-1, D0
      MOVE.L D0, -(A0)

```

conservé } YTRACE : D.W 0
 BINSTR : D.B 0
 BEND : D.B 0
 D.L 0 (sauvegarde)

inhibe Btrace et débogueur

```

      LEA TDEBAS, A0
      MOVE.L (A0)+, (A0)+
      MOVE.L TPPI, (A0)+
      MOVE.L TVALAG, A6
      RTS

```

conservé } TDEBAS pour debug
 TPPI

LEA TPWHILE, A0 } conserve pointeur
 MOVE.L (A0)+, (A0) } pour while ⊗