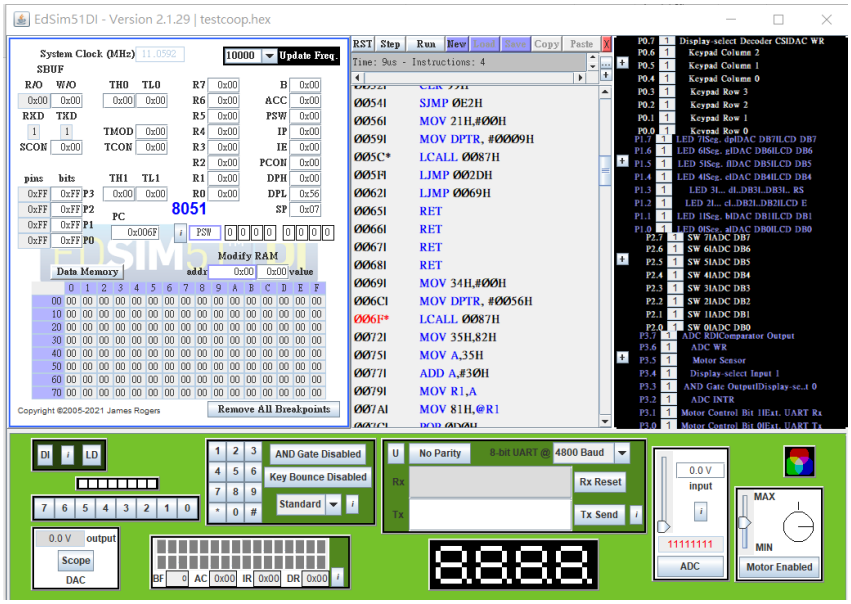
Because I run it on windows system , it seems like the rm instruction in clean doesn't work. So I use del instruction rather than rm.

```
PS C:\Users\Howar\OneDrive\桌面\107034003-ppc1> make clean
del *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym *.asm *.lk
PS C:\Users\Howar\OneDrive\桌面\107034003-ppc1> make
sdcc -c  testcoop.c
testcoop.c:56: warning 158: overflow in implicit constant conversion
sdcc -c  cooperative.c
cooperative.c:149: warning 85: in function ThreadCreate unreferenced function argument : 'fp'
sdcc  -o testcoop.hex testcoop.rel cooperative.rel
```

the address of function

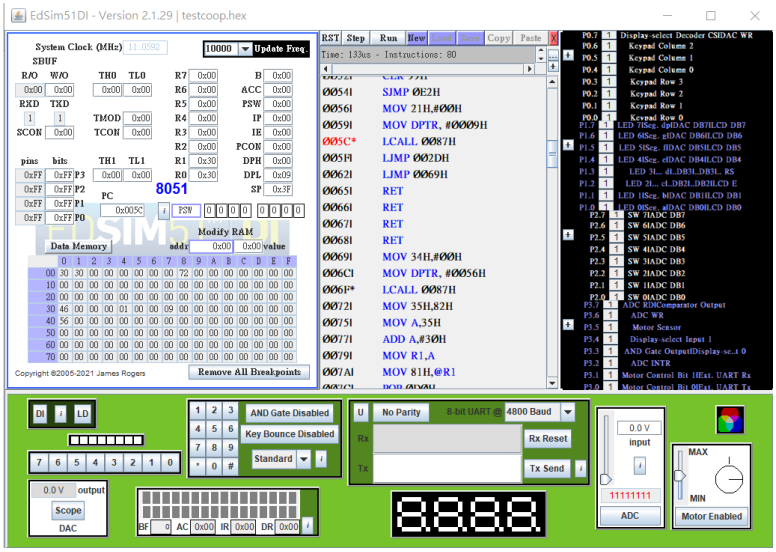
```
GSFINAL                           00000006    00000003 =           3. bytes (REL,CON,CODE)

      Value  Global                            Global Defined In Module
      -----  --------------------------------  ------------------------

^LASxxxx Linker V03.00 + NoICE + sdld,  page 11.
Hexadecimal  [32-Bits]

Area                                  Addr        Size        Decimal Bytes (Attributes)
--------------------------------      ----        ----        ------- ----- ------------
CSEG                              00000009    0000016C =         364. bytes (REL,CON,CODE)

      Value  Global                            Global Defined In Module
      -----  --------------------------------  ------------------------

C:    00000009  _Producer                       testcoop
C:    0000002D  _Consumer                       testcoop
C:    00000056  _main                           testcoop
C:    00000062  __sdcc_gsinit_startup           testcoop
C:    00000066  __mcs51_genRAMCLEAR             testcoop
C:    00000067  __mcs51_genXINIT                testcoop
C:    00000068  __mcs51_genXRAMCLEAR            testcoop
C:    00000069  _Bootstrap                      cooperative
C:    00000087  _ThreadCreate                   cooperative
C:    00000111  _ThreadYield                    cooperative
C:    00000163  _ThreadExit                     cooperative

^LASxxxx Linker V03.00 + NoICE + sdld,  page 12.

Files Linked                          [ module(s) ]
```

bootstrap call threadcreate



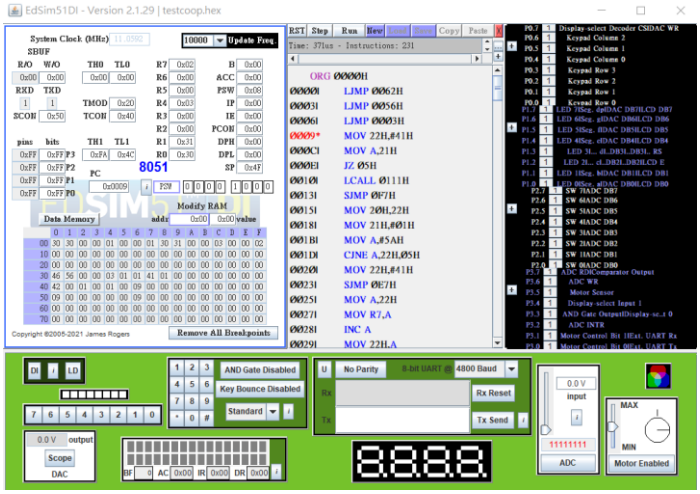

saveSP in 30H-33H is empty, still using the initial stack on 07H,and after executing the instruction LCALL 0087H on 006F, the return address of boostrap will be pushed on 08H,09H.

main call threadcreate



40H-47H not being empty and 46 in 30H, which means thread 0 is active.

producer is running because 0009 is the address of producer function.



consumer is running because 002D is the address of consumer function.