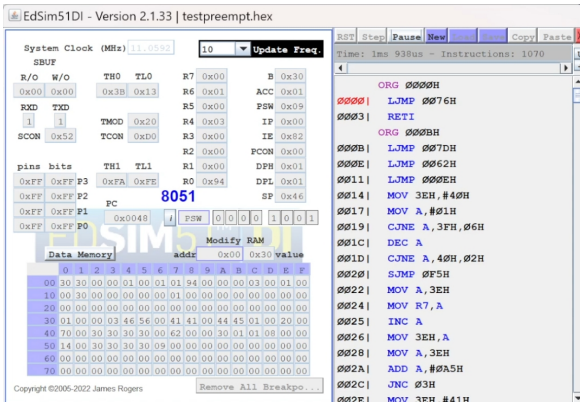
Checkpoint2

1.Typescript for compilation

I chose to implement on the Windows operating system, so I cannot use the 'rm' command. Here, I opted to use the 'del' command to achieve the same functionality.

```
C:\清大資工\OS\112os\ppc2>make clean
del *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym

C:\清大資工\OS\112os\ppc2>make
sdcc -c  testpreempt.c
sdcc -c  preemtive.c
preemtive.c:279: warning 85: in function ThreadCreate unreferenced function argument : 'fp'
sdcc  -o testpreempt.hex testpreempt.rel preemtive.rel
```

2-1.Take one screenshot before each ThreadCreate call. Explain how the stack changes

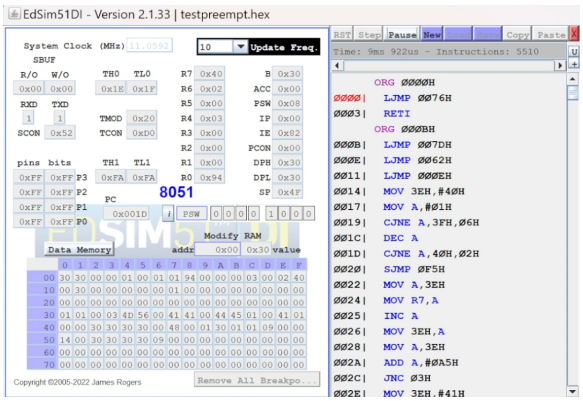


1.The original SP is stored in tempSP (0x38H), and the SP is changed to 0x3F (thread 0 stack).

2.Push DPL DPH in thread0 stack

3.Initial ACC,B,DPL,DPH,PSW value and push those in thread 0 stack,so SP in point to 0x46H

4.Restore original SP from tempSP and Save thread0 SP in 0x34H



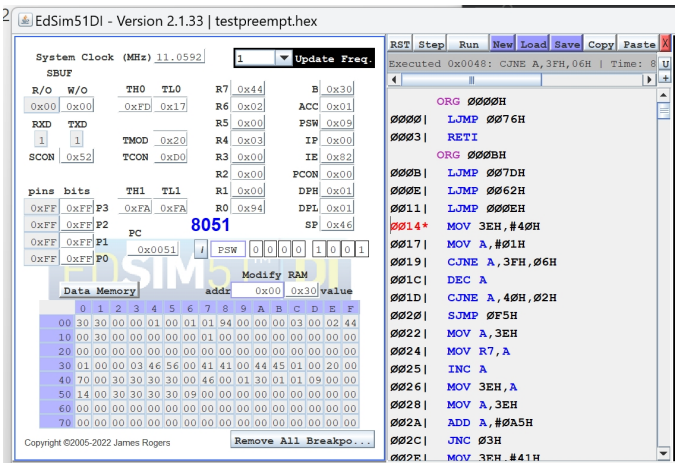
1.The original SP is stored in tempSP (0x38H), and the SP is changed to 0x4F (thread 1 stack).

2.Push DPL DPH in thread 1 stack

3.Initial ACC,B,DPL,DPH,PSW value and push those in thread 1 stack,so SP in point to 0x56H

4.Restore original SP from tempSP and Save thread 1 SP in 0x35H

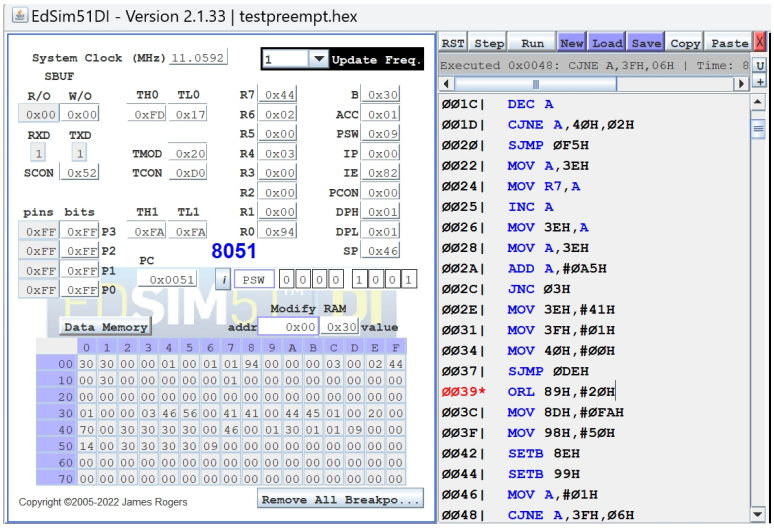
2-2.Take one screenshot when the Producer is running. How do you know?

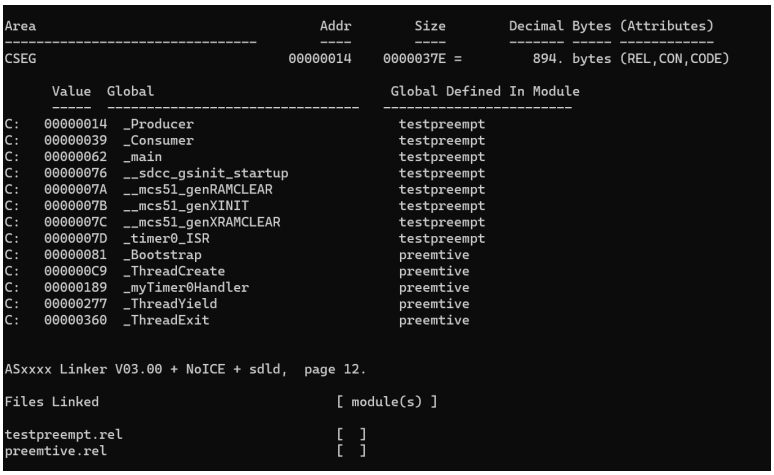


```
Area                                    Addr        Size        Decimal Bytes (Attributes)
--------------------------------   --------   --------   -------- ----- ------------
CSEG                                00000014    0000037E =       894. bytes (REL,CON,CODE)

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

C:   00000014  _Producer                          testpreempt
C:   00000039  _Consumer                          testpreempt
C:   00000062  _main                              testpreempt
C:   00000076  __sdcc_gsinit_startup              testpreempt
C:   0000007A  __mcs51_genRAMCLEAR                testpreempt
C:   0000007B  __mcs51_genXINIT                   testpreempt
C:   0000007C  __mcs51_genXRAMCLEAR               testpreempt
C:   0000007D  _timer0_ISR                        testpreempt
C:   00000081  _Bootstrap                         preemtive
C:   000000C9  _ThreadCreate                      preemtive
C:   00000189  _myTimer0Handler                   preemtive
C:   00000277  _ThreadYield                       preemtive
C:   00000360  _ThreadExit                        preemtive


ASxxxx Linker V03.00 + NoICE + sdld,  page 12.

Files Linked                            [ module(s) ]

testpreempt.rel                         [ ]
preemtive.rel                           [ ]
```

producer is running,because 0014 is the address of producer function.

2-3.Take one screenshot when the Consumer is running. How do you know?





```
Area                                    Addr        Size        Decimal Bytes (Attributes)
--------------------------------        ----        ----        ------- ----- ------------
CSEG                                 00000014    0000037E =         894. bytes (REL,CON,CODE)

      Value  Global                             Global Defined In Module
      -----  --------------------------------   ------------------------
C:   00000014  _Producer                          testpreempt
C:   00000039  _Consumer                          testpreempt
C:   00000062  _main                              testpreempt
C:   00000076  __sdcc_gsinit_startup              testpreempt
C:   0000007A  __mcs51_genRAMCLEAR                testpreempt
C:   0000007B  __mcs51_genXINIT                   testpreempt
C:   0000007C  __mcs51_genXRAMCLEAR               testpreempt
C:   0000007D  _timer0_ISR                        testpreempt
C:   00000081  _Bootstrap                         preemtive
C:   000000C9  _ThreadCreate                      preemtive
C:   00000189  _myTimer0Handler                   preemtive
C:   00000277  _ThreadYield                       preemtive
C:   00000360  _ThreadExit                        preemtive


ASxxxx Linker V03.00 + NoICE + sdld,  page 12.

Files Linked                            [ module(s) ]

testpreempt.rel                         [  ]
preemtive.rel                           [  ]
```

consumer is running,because 0039 is the address of consumer function.

2-4.How can you tell that the interrupt is triggering on a regular basis?

Due to the PC address being switched between 0x17, 0x19(Producer’s addr.), and 0x46, 0x48, 0x51(Consumer’s addr.), interrupts are triggered on a regular basis.