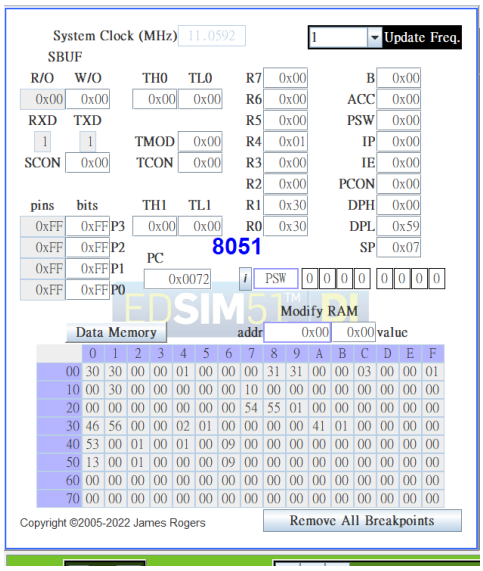
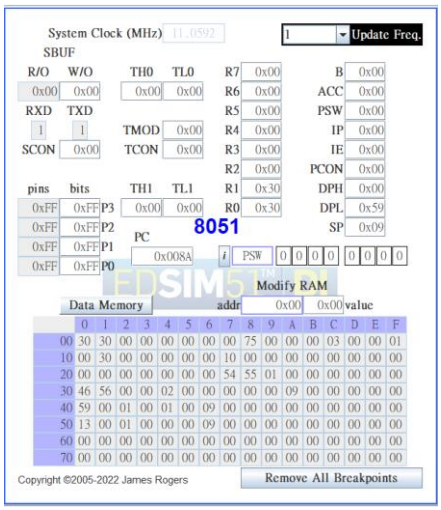
109062119 李佳栩

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

(1)

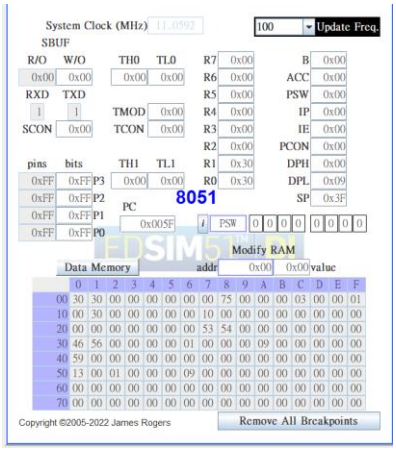


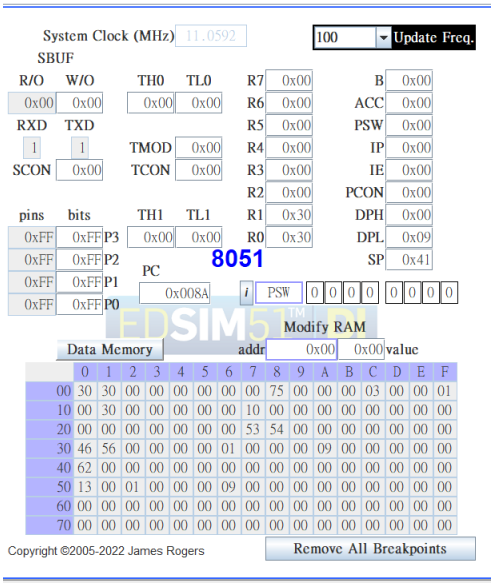


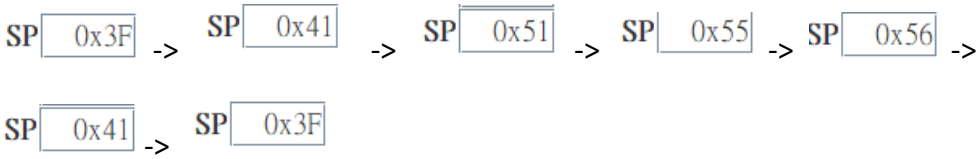
The initial stack is in 0x07, and after call LCALL it saves the return address in stack, so it becomes 0x09. Then, the stack thread 1 use start from 0x40 to 0x4F. After push DPL, DPH, the stack is in 0x41. Later it push 0 into stack 4 times and PSW. In the end, it return to 0x09 to get the return address.

P 0x3F -> SP 0x41 -> SP 0x45 -> SP 0x46 -> SP 0x09 -> SP 0x07

(2)

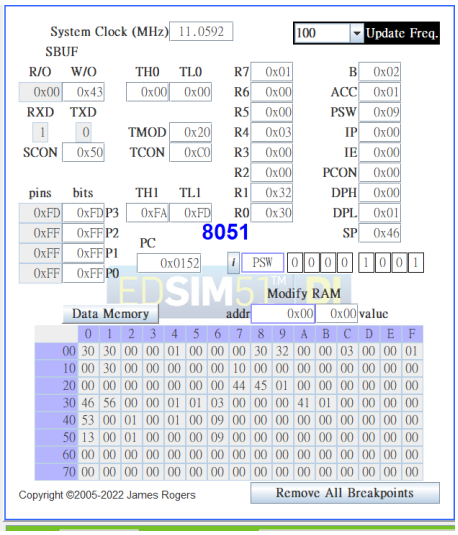




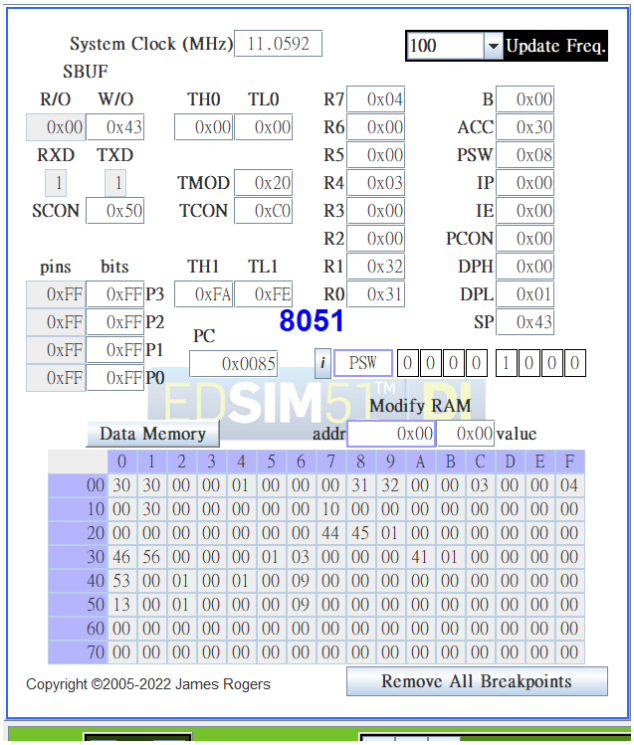


The second threadcreate is called in main() for procuder. The initial stack is in 0x3F, and after call LCALL it saves the return address in stack, so it becomes 0x41. Then, the stack thread 1 use start from 0x50 to 0x5F. After push DPL, DPH, the stack is in 0x51. Later it push 0 into stack 4 times and PSW. In the end, it return to 0x41 to get the return address.

- Take one screenshot when the `Producer` is running. How do you know? Since my cur_thread is in 0x34, when 0x34 is 1, it means that Producer is running.



- Take one screenshot when the `Consumer` is running. How do you know? Since my cur_thread is in 0x34, when 0x34 is 0, it means that Producer is running.



Typescript

Make clean other files, and make execute sdcc -c and execute sdcc -o to output a file testcoop.hex to be loaded into edsim.

```
PS C:\Users\josh9\OneDrive\桌面\大三上\OS\checkpoint1> make clean
del *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym
PS C:\Users\josh9\OneDrive\桌面\大三上\OS\checkpoint1> 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'
cooperative.c:240: warning 158: overflow in implicit constant conversion
sdcc  -o testcoop.hex testcoop.rel cooperative.rel
PS C:\Users\josh9\OneDrive\桌面\大三上\OS\checkpoint1>
```