Student ID : 111062307, Name : 陳大佑

```
__data __at (0x30) char saved_SP[MAXTHREADS];
__data __at (0x34) ThreadID cur_ID;
__data __at (0x35) char bitmap_ID;
__data __at (0x36) char temp_SP;        // temporary save the SP
__data __at (0x37) ThreadID new_ID;
```

```
__data __at (0x38) char sharedBuffer;
__data __at (0x39) char bufferFull;     // 0: empty, 1 : full
__data __at (0x3A) char nextChar;
```

Set some parameters on manually allocated memory.

```
void myTimer0Handler(void) {
    __critical {
        SAVESTATE;
        if (!bitmap_ID) ;
        else if (cur_ID == 0) {
            if (bitmap_ID & 0x2) cur_ID = 1;
            else if (bitmap_ID & 0x4) cur_ID = 2;
            else if (bitmap_ID & 0x8) cur_ID = 3;
        } else if (cur_ID == 1) {
            if (bitmap_ID & 0x4) cur_ID = 2;
            else if (bitmap_ID & 0x8) cur_ID = 3;
            else if (bitmap_ID & 0x1) cur_ID = 0;
        } else if (cur_ID == 2) {
            if (bitmap_ID & 0x8) cur_ID = 3;
            else if (bitmap_ID & 0x1) cur_ID = 0;
            else if (bitmap_ID & 0x2) cur_ID = 1;
        } else if (cur_ID == 3) {
            if (bitmap_ID & 0x1) cur_ID = 0;
            else if (bitmap_ID & 0x2) cur_ID = 1;
            else if (bitmap_ID & 0x4) cur_ID = 2;
        }

        RESTORESTATE;
    }
    __asm
        RETI
    __endasm;
}
```

If bitmap_ID can’t match anything then do nothing.

If cur_ID is 0 then check whether exists other thread or not.

If so, switch to it.

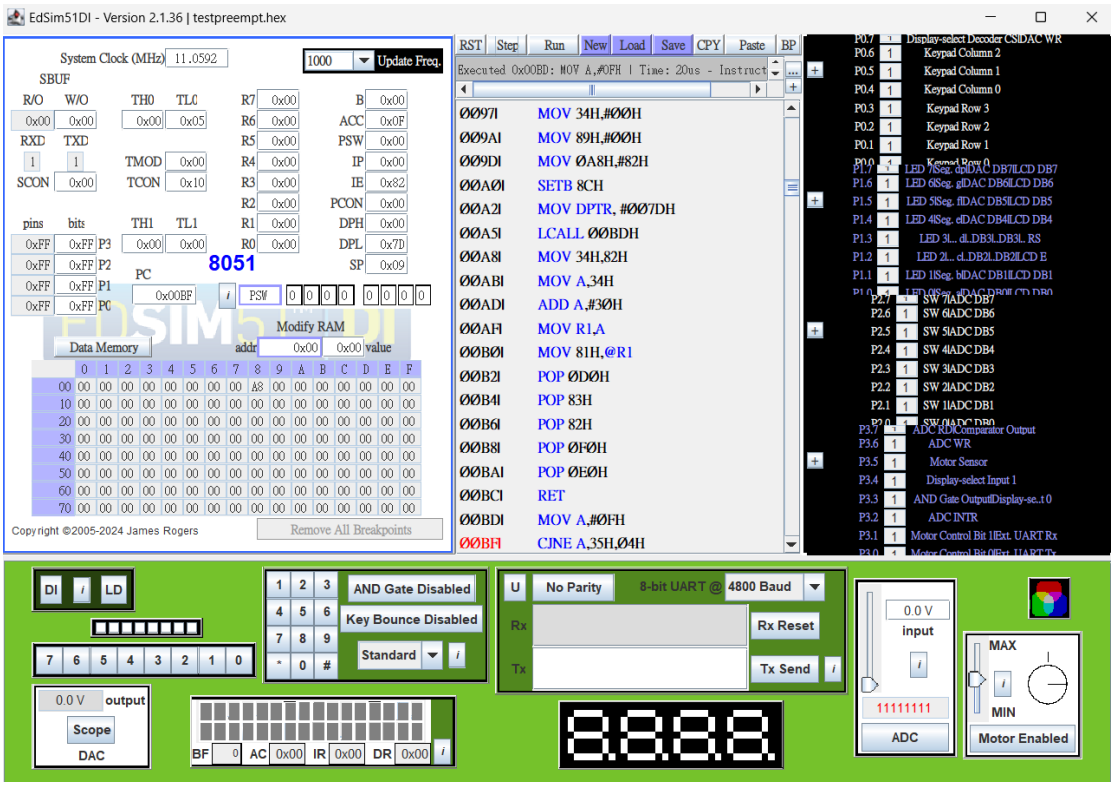
And if cur_ID is 1 or 2 or 3 do same thing like cur_ID = 0.

Screenshots for compliation.

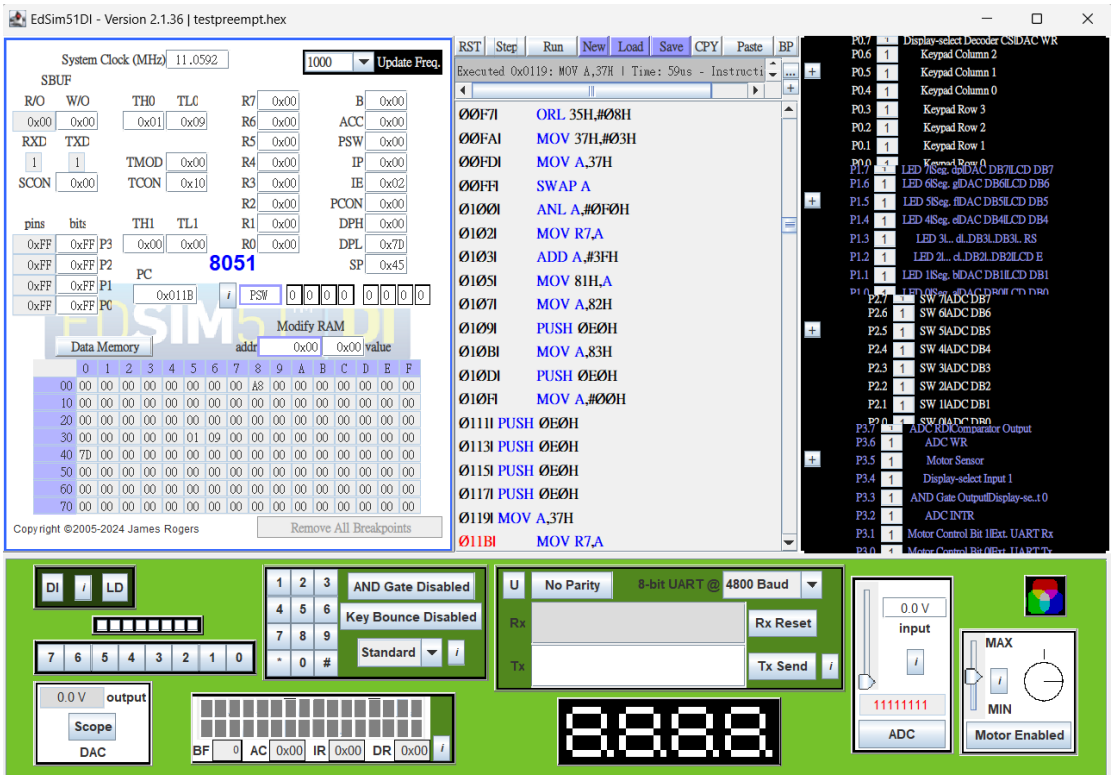
```
dylan@LAPTOP-S5FOLV5V ~/os/ppc2
$ make clean
rm *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym *.asm *.lk
rm: cannot remove '*.ihx': No such file or directory
rm: cannot remove '*.lnk': No such file or directory
make: *** [clean] Error 1

dylan@LAPTOP-S5FOLV5V ~/os/ppc2
$ make
sdcc -c  testpreempt.c
sdcc -c  preemptive.c
preemptive.c:161: warning 85: in function ThreadCreate unreferenced function argument : 'fp'
sdcc  -o testpreempt.hex testpreempt.rel preemptive.rel
```

Screenshots and explanation:

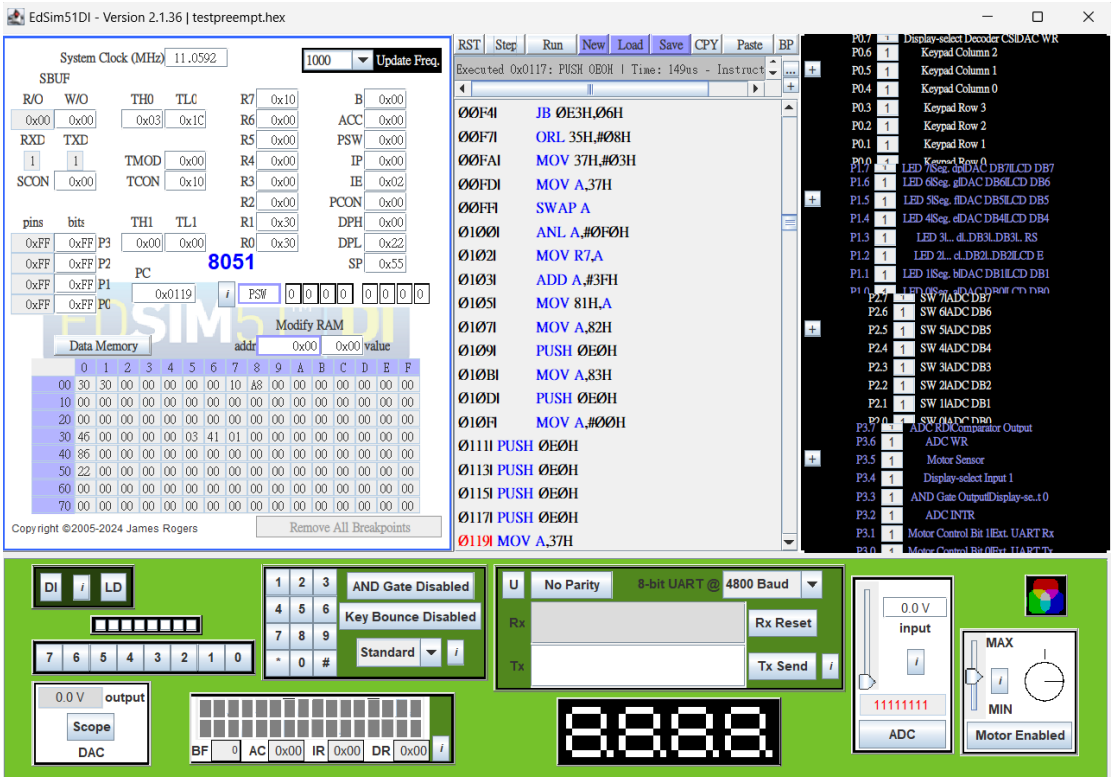
Before the main thread was created, since by default the SP = 0x07, the 0x08 was numbered by first instruction(bootstrap).



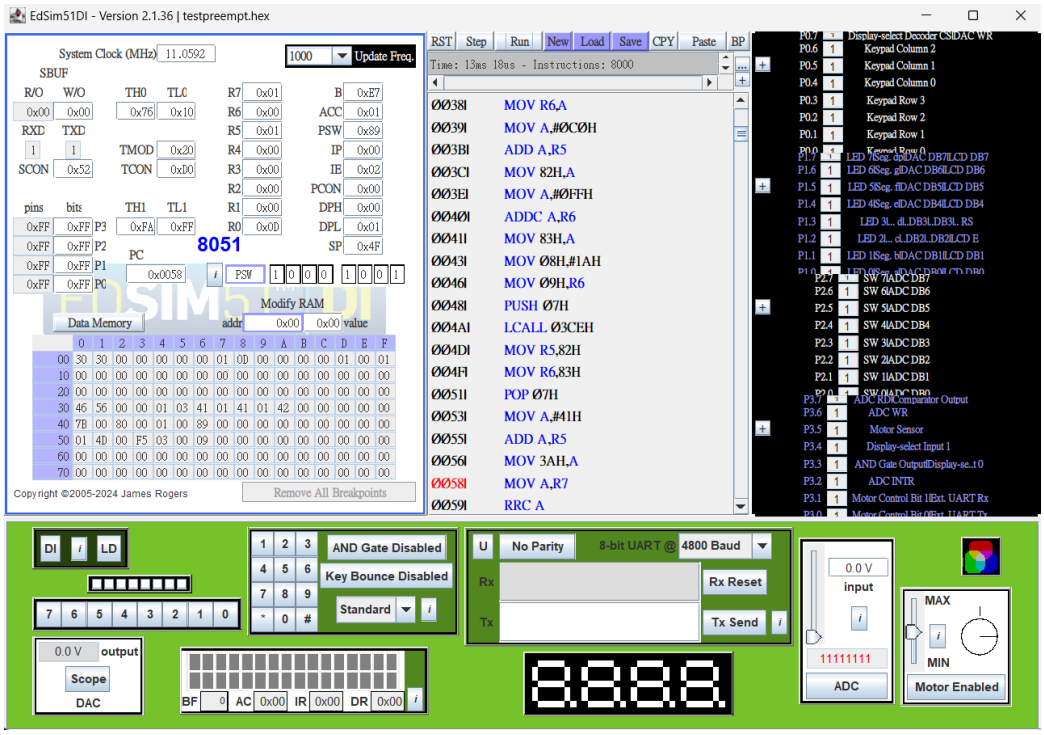
After main thread had been created, the address of 40H was stored for main's address, and 42H, 43H, 44H, 45H, 46H are ACC, B, DPL, DPH, PSW, respectively.



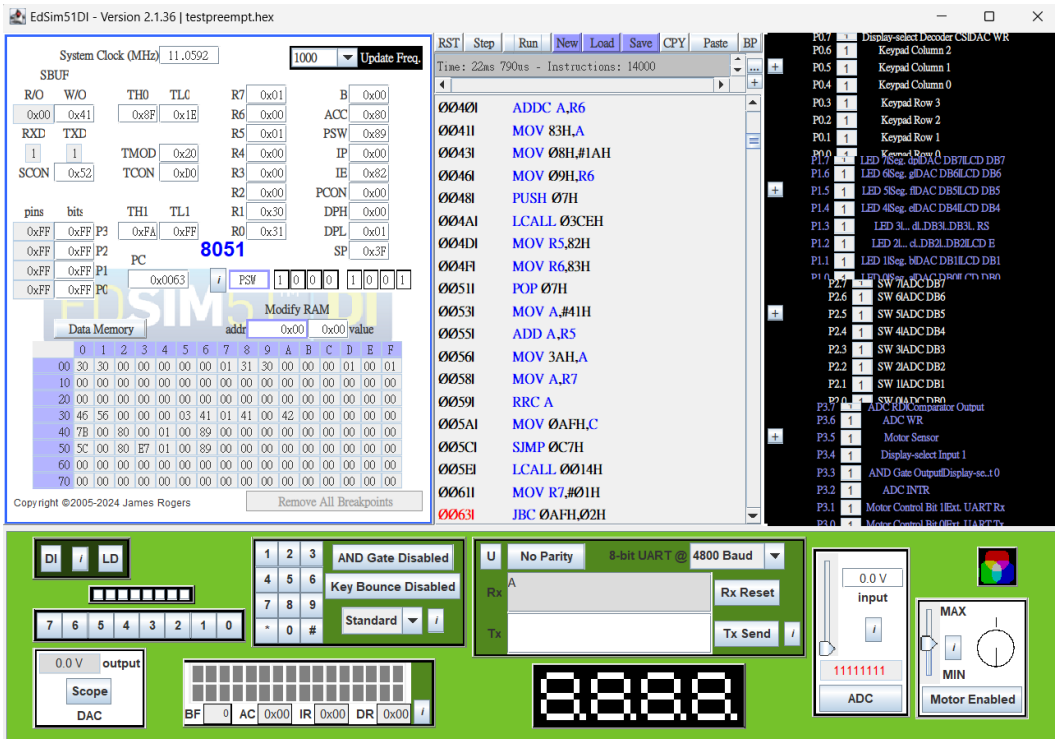
And after producer was created, the address of the 50H was stored for producer's address. and 52H~56H are ACC, B, DPL, DPH, PSW, respectively.



We can know the producer is running, since cur_ID on address(34H) is 01 which means producer and compare to previous picture we can find the addresses of(38H, 39H, 3AH) are replaced by other number(char'A').

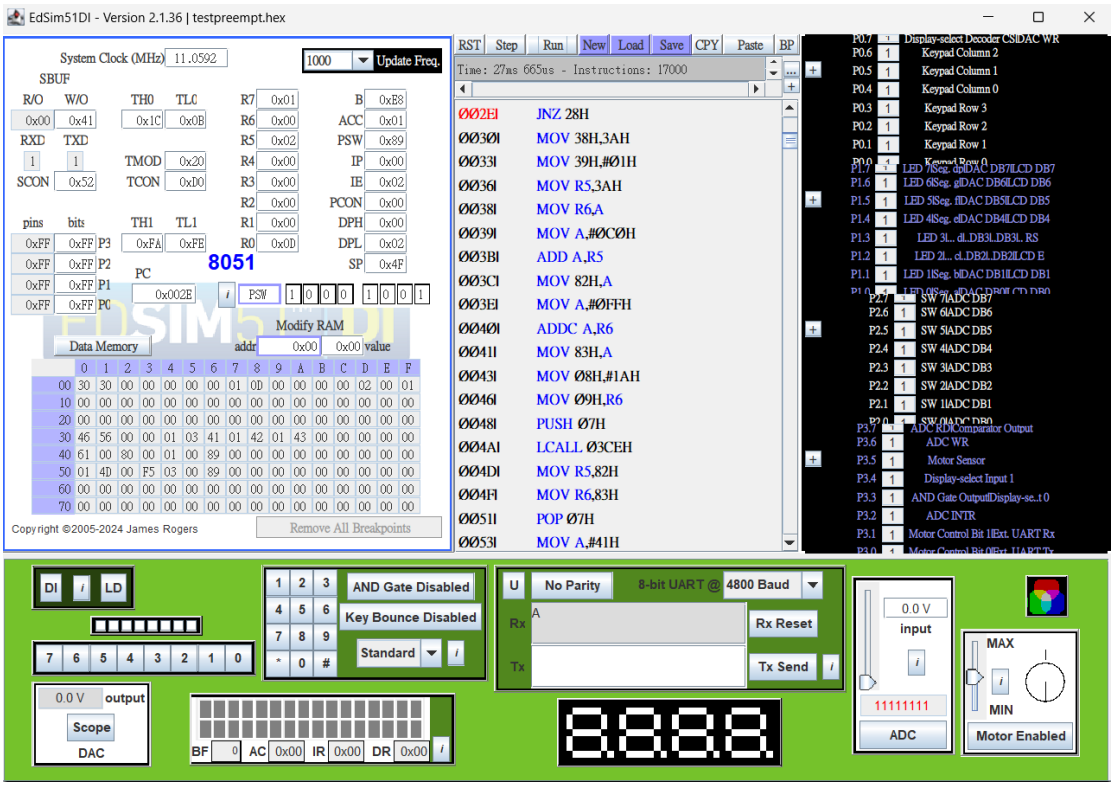


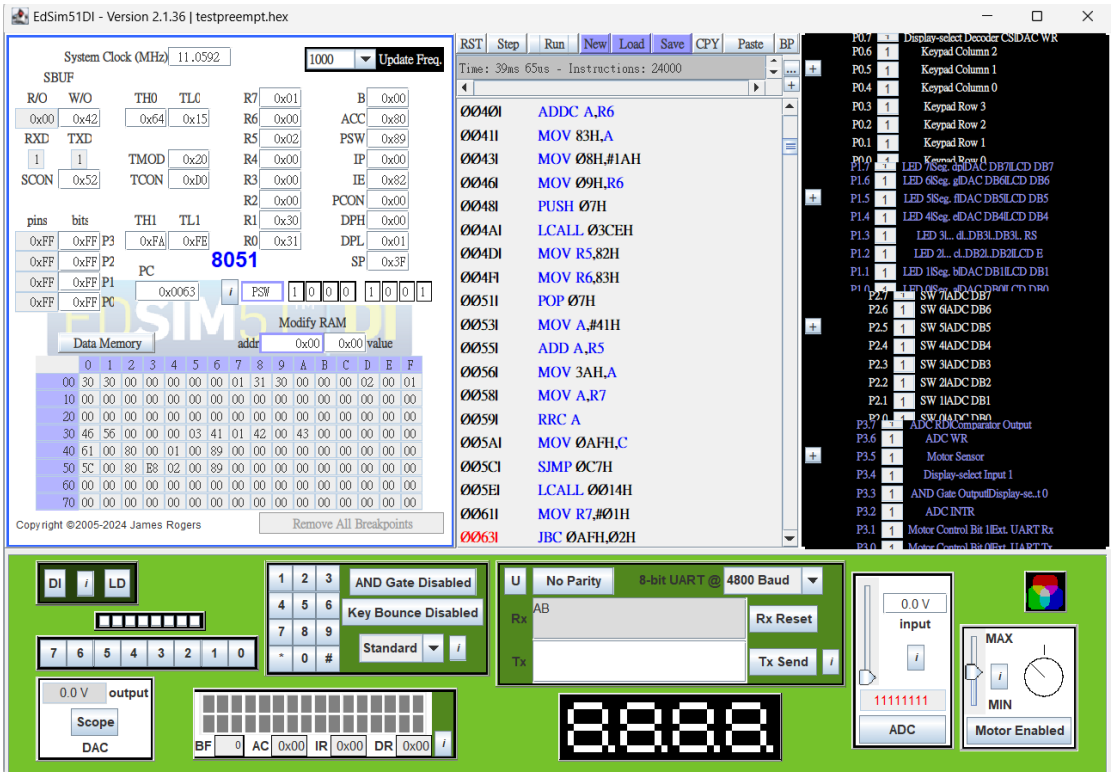
We can know the consumer is running, since cur_ID on address(34H) is 00 which means it switches. And the address(39H) is changed from 1 to 0.

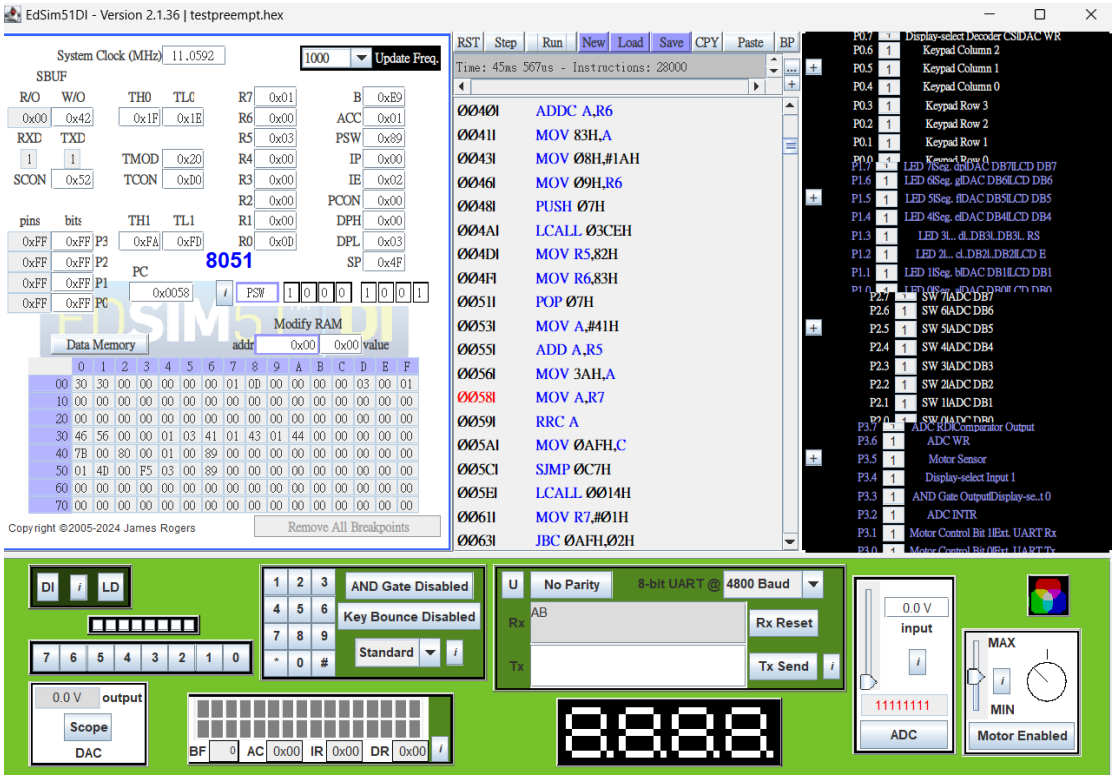


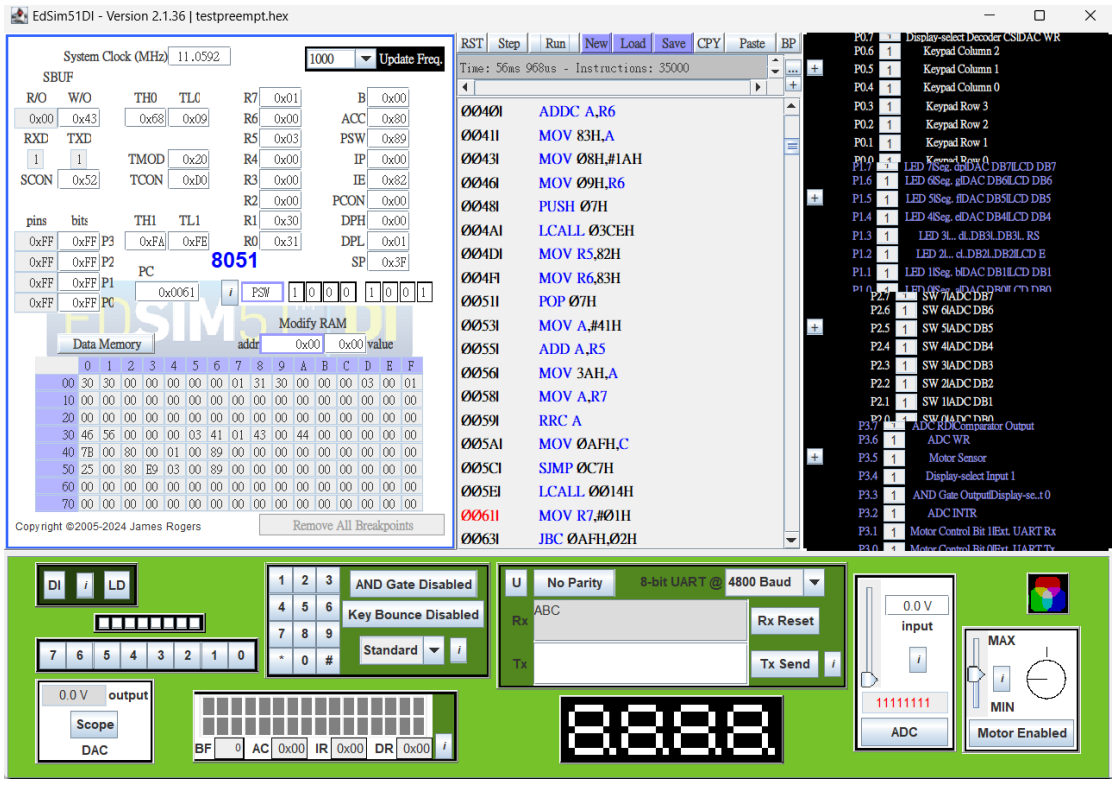
We can see the running time of the producer is at 45, 28, and 13,

and the running time of consumer is at 56, 39, and 23, which difference is regular about 10. So interrupt is triggering on a regular basis.






EdSim51DI - Version 2.1.36 | testpreempt.hex
System Clock (MHz) 11.0592
1000
Update Freq.
8051
PC
0x0058
Modify RAM
Data Memory
Copyright ©2005-2024 James Rogers
Remove All Breakpoints
RST
Step
Run
New
Load
Save
CPY
Paste
BP
Time: 45ms 567us - Instructions: 28000
ØØ4ØH ADDC A,R6
ØØ41H MOV 83H,A
ØØ43H MOV Ø8H,#1AH
ØØ46H MOV Ø9H,R6
ØØ48H PUSH Ø7H
ØØ4AH LCALL Ø3CEH
ØØ4DH MOV R5,82H
ØØ4FH MOV R6,83H
ØØ51H POP Ø7H
ØØ53H MOV A,#41H
ØØ55H ADD A,R5
ØØ56H MOV 3AH,A
ØØ58H MOV A,R7
ØØ59H RRC A
ØØ5AH MOV ØAFH,C
ØØ5CH SJMP ØC7H
ØØ5EH LCALL ØØ14H
ØØ61H MOV R7,#Ø1H
ØØ63H JBC ØAFH,Ø2H
AND Gate Disabled
Key Bounce Disabled
Standard
No Parity
8-bit UART @
4800 Baud
Rx Reset
Tx Send
Motor Enabled
MAX
MIN
ADC
Scope
DAC


EdSim51DI - Version 2.1.36 | testpreempt.hex
System Clock (MHz) 11.0592
1000
Update Freq.
8051
PC
0x0061
Modify RAM
Data Memory
Copyright ©2005-2024 James Rogers
Remove All Breakpoints
RST
Step
Run
New
Load
Save
CPY
Paste
BP
Time: 56ms 968us - Instructions: 35000
ØØ4ØH ADDC A,R6
ØØ41H MOV 83H,A
ØØ43H MOV Ø8H,#1AH
ØØ46H MOV Ø9H,R6
ØØ48H PUSH Ø7H
ØØ4AH LCALL Ø3CEH
ØØ4DH MOV R5,82H
ØØ4FH MOV R6,83H
ØØ51H POP Ø7H
ØØ53H MOV A,#41H
ØØ55H ADD A,R5
ØØ56H MOV 3AH,A
ØØ58H MOV A,R7
ØØ59H RRC A
ØØ5AH MOV ØAFH,C
ØØ5CH SJMP ØC7H
ØØ5EH LCALL ØØ14H
ØØ61H MOV R7,#Ø1H
ØØ63H JBC ØAFH,Ø2H
AND Gate Disabled
Key Bounce Disabled
Standard
No Parity
8-bit UART @
4800 Baud
Rx Reset
Tx Send
Motor Enabled
MAX
MIN
ADC
Scope
DAC