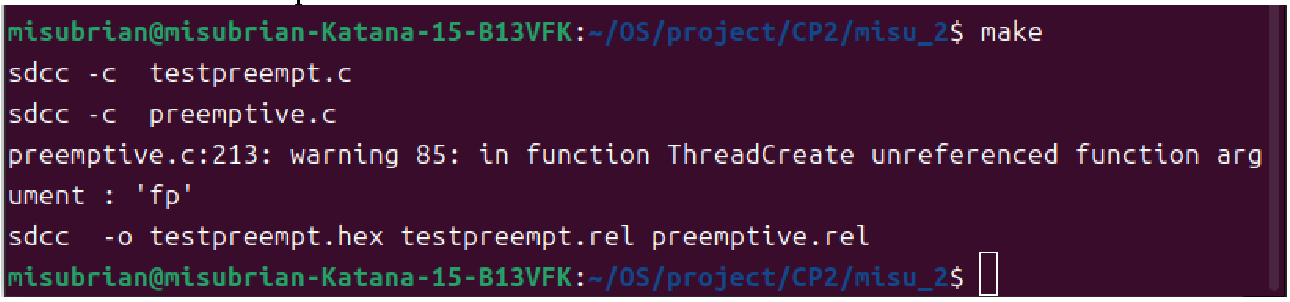
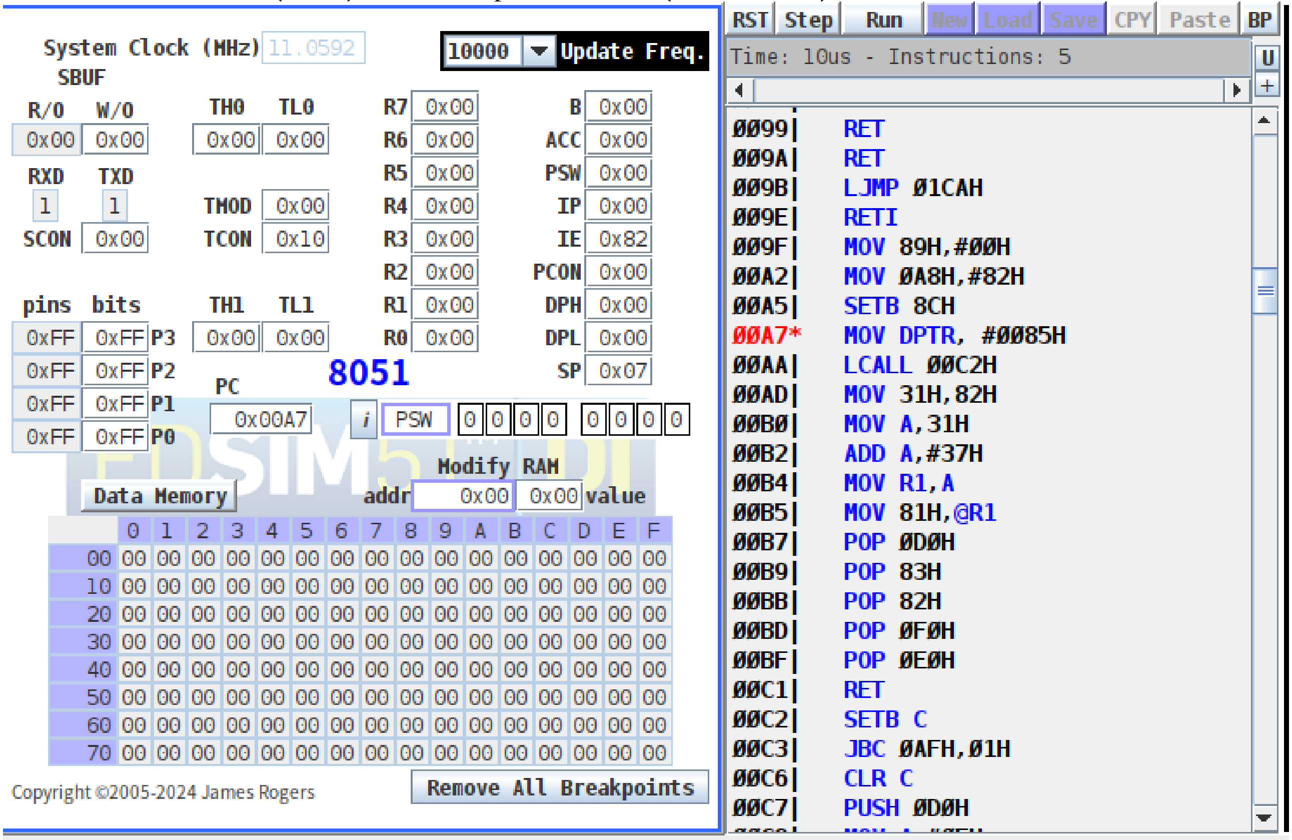
Report of Project Checkpoint 2
112062630 JING-YONG, SU 蘇敬詠

1. Screenshots for compilation



```
misubrian@misubrian-Katana-15-B13VFK:~/OS/project/CP2/misu_2$ make
sdcc -c  testpreempt.c
sdcc -c  preemptive.c
preemptive.c:213: warning 85: in function ThreadCreate unreferenced function argument : 'fp'
sdcc  -o testpreempt.hex testpreempt.rel preemptive.rel
misubrian@misubrian-Katana-15-B13VFK:~/OS/project/CP2/misu_2$ 
```

2. (a)
(i)
Before ThreadCreate(main), the stack point is 0x07(initial stack).

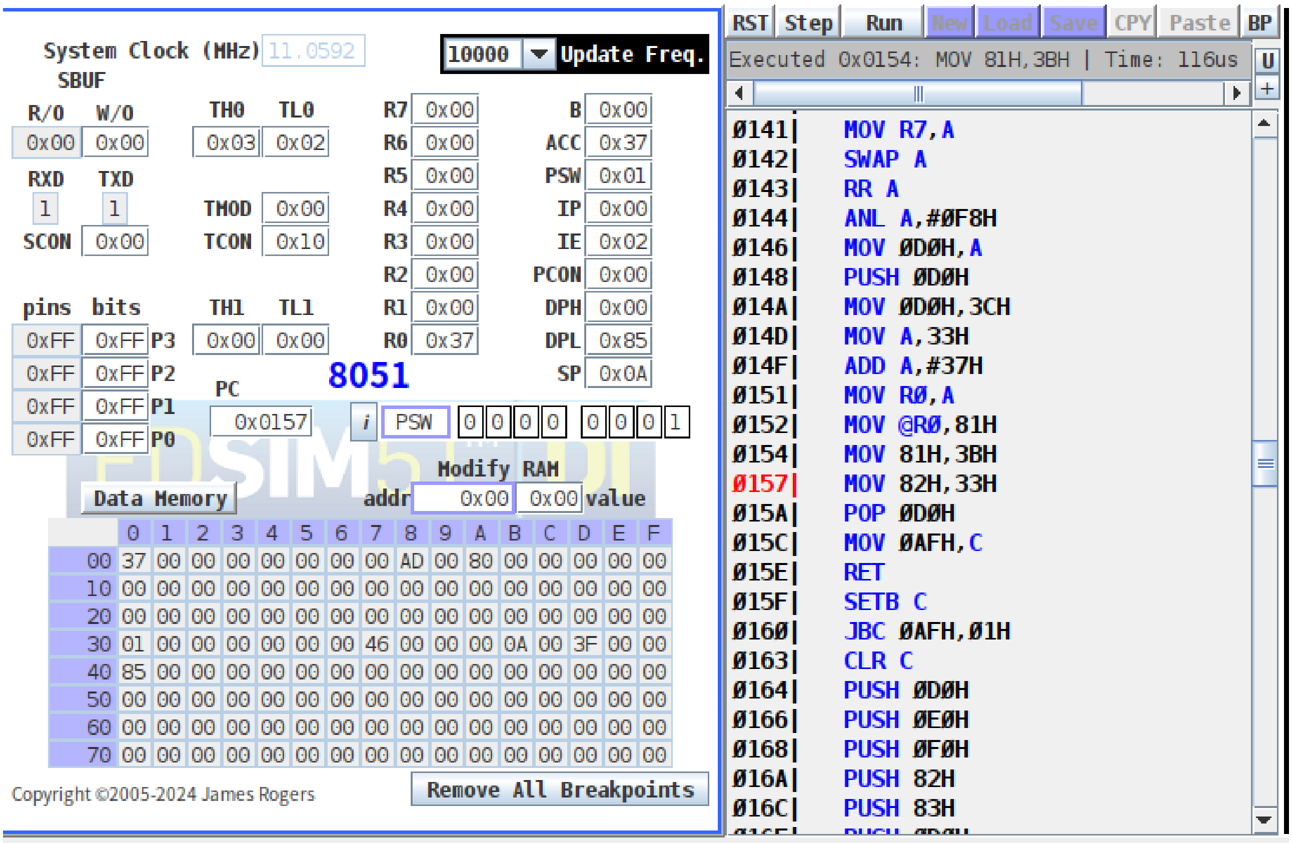


```
System Clock (MHz) 11.0592      10000
SBUF
R/O 0x00   W/O 0x00
RXD 1      TXD 1
SCON 0x00
TH0 0x00   TL0 0x00
TMOD 0x00
TCON 0x10
TH1 0x00   TL1 0x00
pins  bits
0xFF  0xFF P3
0xFF  0xFF P2
0xFF  0xFF P1
0xFF  0xFF P0
R7 0x00    B 0x00
R6 0x00    ACC 0x00
R5 0x00    PSW 0x00
R4 0x00    IP 0x00
R3 0x00    IE 0x82
R2 0x00    PCON 0x00
R1 0x00    DPH 0x00
R0 0x00    DPL 0x00
           SP 0x07
PC 0x00A7
PSW 0 0 0 0 0 0 0 0
addr 0x00  0x00 value
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

```
Time: 10us - Instructions: 5

0099|   RET
009A|   RET
009B|   LJMP 01CAH
009E|   RETI
009F|   MOV 89H,#00H
00A2|   MOV 0A8H,#82H
00A5|   SETB 8CH
00A7*   MOV DPTR, #0085H
00AA|   LCALL 00C2H
00AD|   MOV 31H,82H
00B0|   MOV A,31H
00B2|   ADD A,#37H
00B4|   MOV R1,A
00B5|   MOV 81H,@R1
00B7|   POP 0D0H
00B9|   POP 83H
00BB|   POP 82H
00BD|   POP 0F0H
00BF|   POP 0E0H
00C1|   RET
00C2|   SETB C
00C3|   JBC 0AFH,01H
00C6|   CLR C
00C7|   PUSH 0D0H
```

When ThreadCreate(main) is running, the stack point become 0x09 due to the return address is pushed.
After finishing creation of main, the stack point 0x46 is saved to savedSP[0] (i.e. 0x37). Then, the stack is recovered to initial stack.



System Clock (MHz) 11.0592 — 10000

SBUF R/O	SBUF W/O	TH0	TL0	TMOD	TCON	TH1	TL1
0x00	0x00	0x03	0x02	0x00	0x10	0x00	0x00

RXD 1 TXD 1 SCON 0x00

	pins	bits
P3	0xFF	0xFF
P2	0xFF	0xFF
P1	0xFF	0xFF
P0	0xFF	0xFF

R7	R6	R5	R4	R3	R2	R1	R0
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x37

B	ACC	PSW	IP	IE	PCON	DPH	DPL	SP
0x00	0x37	0x01	0x00	0x02	0x00	0x00	0x85	0x0A

PC 0x0157

PSW 0 0 0 0 0 0 0 1

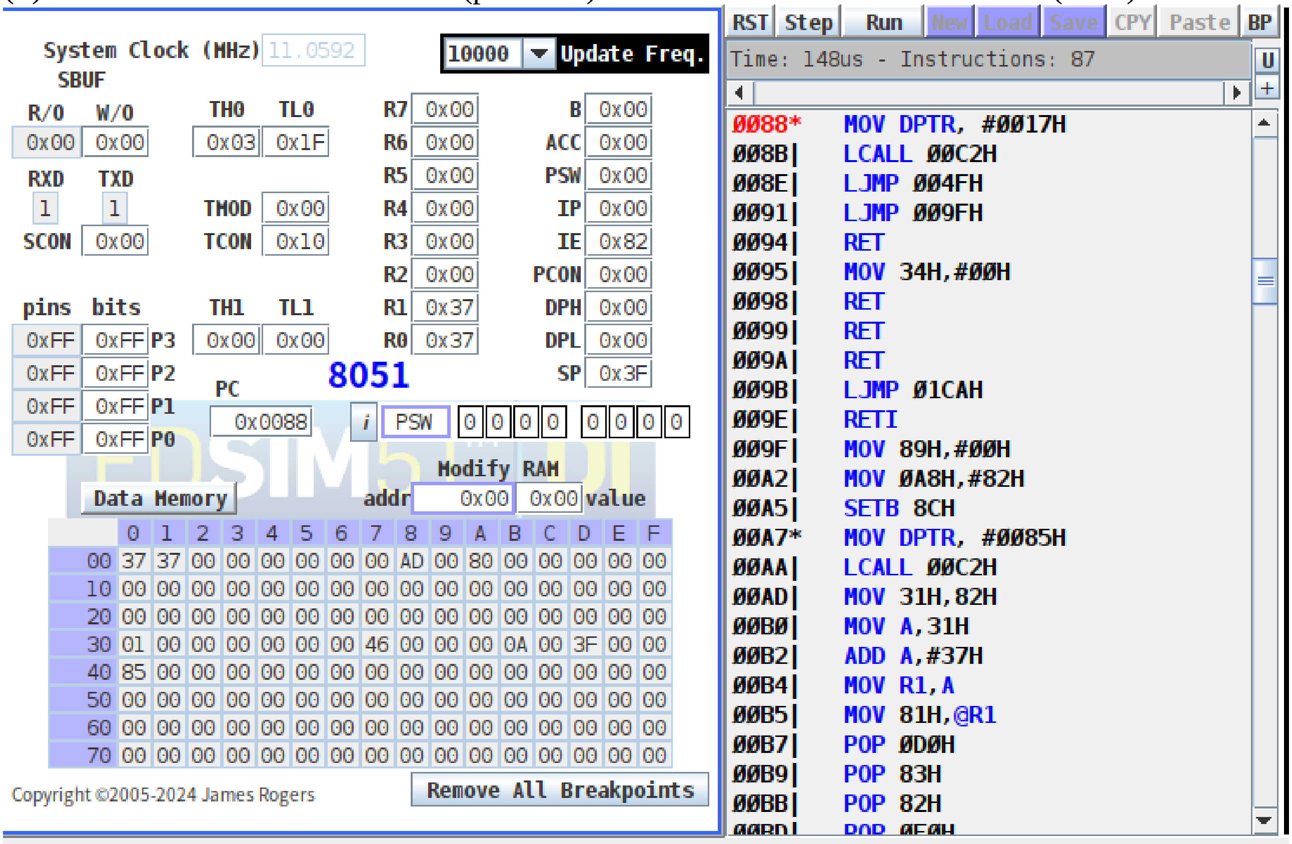
addr 0x00 0x00 value

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	37	00	00	00	00	00	00	00	AD	00	80	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	01	00	00	00	00	00	00	46	00	00	00	0A	00	3F	00	00
40	85	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Executed 0x0154: MOV 81H,3BH | Time: 116us

```
Ø141|   MOV R7,A
Ø142|   SWAP A
Ø143|   RR A
Ø144|   ANL A,#ØF8H
Ø146|   MOV ØDØH,A
Ø148|   PUSH ØDØH
Ø14A|   MOV ØDØH,3CH
Ø14D|   MOV A,33H
Ø14F|   ADD A,#37H
Ø151|   MOV RØ,A
Ø152|   MOV @RØ,81H
Ø154|   MOV 81H,3BH
Ø157|   MOV 82H,33H
Ø15A|   POP ØDØH
Ø15C|   MOV ØAFH,C
Ø15E|   RET
Ø15F|   SETB C
Ø160|   JBC ØAFH,Ø1H
Ø163|   CLR C
Ø164|   PUSH ØDØH
Ø166|   PUSH ØEØH
Ø168|   PUSH ØFØH
Ø16A|   PUSH 82H
Ø16C|   PUSH 83H
Ø16E|   PUSH ØDØH
```

(ii) Screenshot before ThreadCreate(producer). SP = 0x3F is the stack of thread 0 (main).



System Clock (MHz) 11.0592 — 10000

SBUF R/O	SBUF W/O	TH0	TL0	TMOD	TCON	TH1	TL1
0x00	0x00	0x03	0x1F	0x00	0x10	0x00	0x00

RXD 1 TXD 1 SCON 0x00

	pins	bits
P3	0xFF	0xFF
P2	0xFF	0xFF
P1	0xFF	0xFF
P0	0xFF	0xFF

R7	R6	R5	R4	R3	R2	R1	R0
0x00	0x00	0x00	0x00	0x00	0x00	0x37	0x37

B	ACC	PSW	IP	IE	PCON	DPH	DPL	SP
0x00	0x00	0x00	0x00	0x82	0x00	0x00	0x00	0x3F

PC 0x0088

PSW 0 0 0 0 0 0 0 0

addr 0x00 0x00 value

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	37	37	00	00	00	00	00	00	AD	00	80	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	01	00	00	00	00	00	00	46	00	00	00	0A	00	3F	00	00
40	85	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

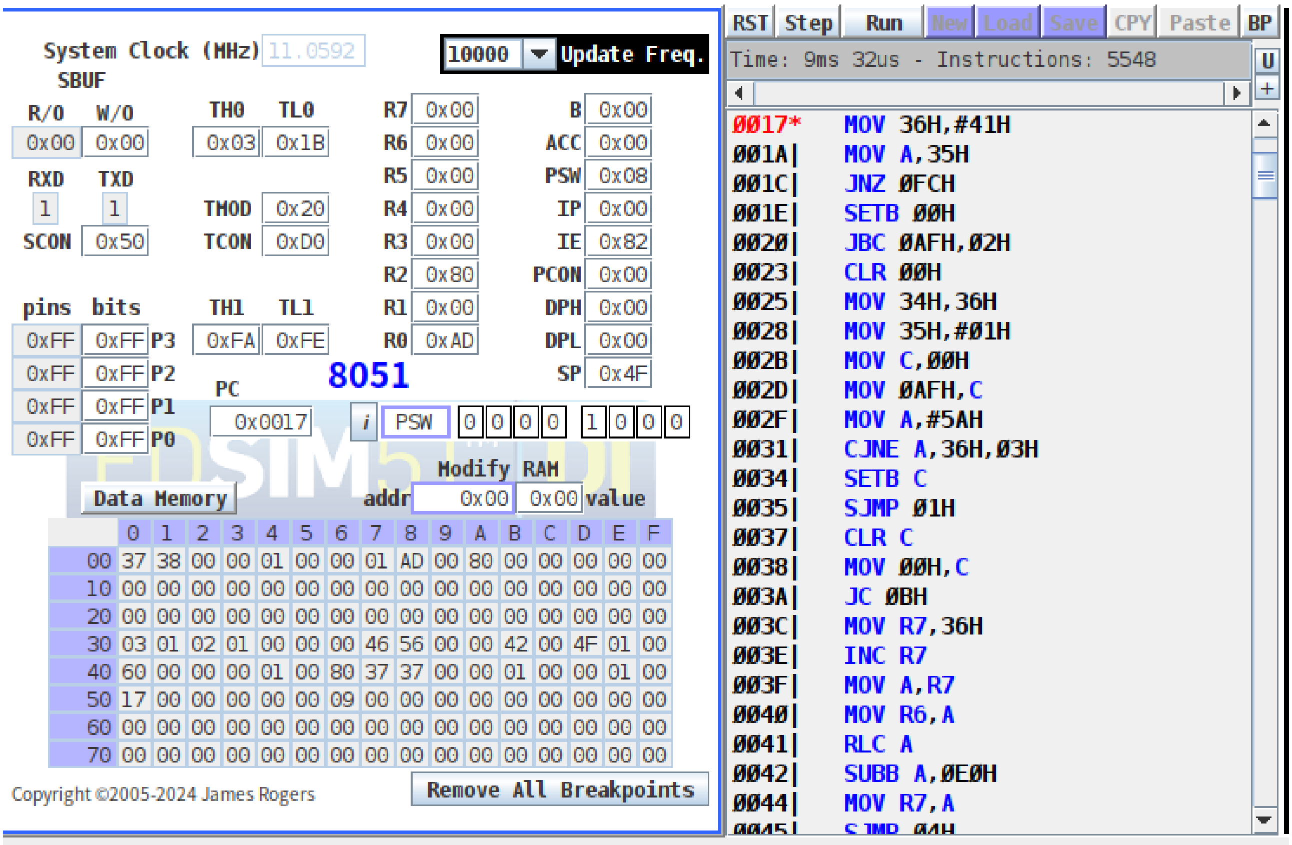
Time: 148us - Instructions: 87

```
ØØ88*   MOV DPTR, #ØØ17H
ØØ8B|   LCALL ØØC2H
ØØ8E|   LJMP ØØ4FH
ØØ91|   LJMP ØØ9FH
ØØ94|   RET
ØØ95|   MOV 34H,#ØØH
ØØ98|   RET
ØØ99|   RET
ØØ9A|   RET
ØØ9B|   LJMP Ø1CAH
ØØ9E|   RETI
ØØ9F|   MOV 89H,#ØØH
ØØA2|   MOV ØA8H,#82H
ØØA5|   SETB 8CH
ØØA7*   MOV DPTR, #ØØ85H
ØØAA|   LCALL ØØC2H
ØØAD|   MOV 31H,82H
ØØBØ|   MOV A,31H
ØØB2|   ADD A,#37H
ØØB4|   MOV R1,A
ØØB5|   MOV 81H,@R1
ØØB7|   POP ØDØH
ØØB9|   POP 83H
ØØBB|   POP 82H
ØØBD|   POP ØEØH
```

SP becomes 0x42 later due to pushing return address.
Then, jump to stack of thread 1 at 0x4F in order to create producer.
Finally, stack point of producer (0x56) is saved to 0x38 and is recovered to SP of main (0x42).

2. (b)

Producer is running, due to <PSW.4, PSW.3> = 01.



System Clock (MHz) 11.0592

10000 Update Freq.

SBUF R/O 0x00 W/O 0x00

RXD 1 TXD 1

SCON 0x50

TH0 0x03 TL0 0x1B

TMOD 0x20 TCON 0xD0

TH1 0xFA TL1 0xFE

R7 0x00
R6 0x00
R5 0x00
R4 0x00
R3 0x00
R2 0x80
R1 0x00
R0 0xAD

B 0x00
ACC 0x00
PSW 0x08
IP 0x00
IE 0x82
PCON 0x00
DPH 0x00
DPL 0x00
SP 0x4F

pins	bits	
0xFF	0xFF	P3
0xFF	0xFF	P2
0xFF	0xFF	P1
0xFF	0xFF	P0

8051

PC 0x0017

PSW 0 0 0 0 1 0 0 0

Modify RAM addr 0x00 value 0x00

Data Memory

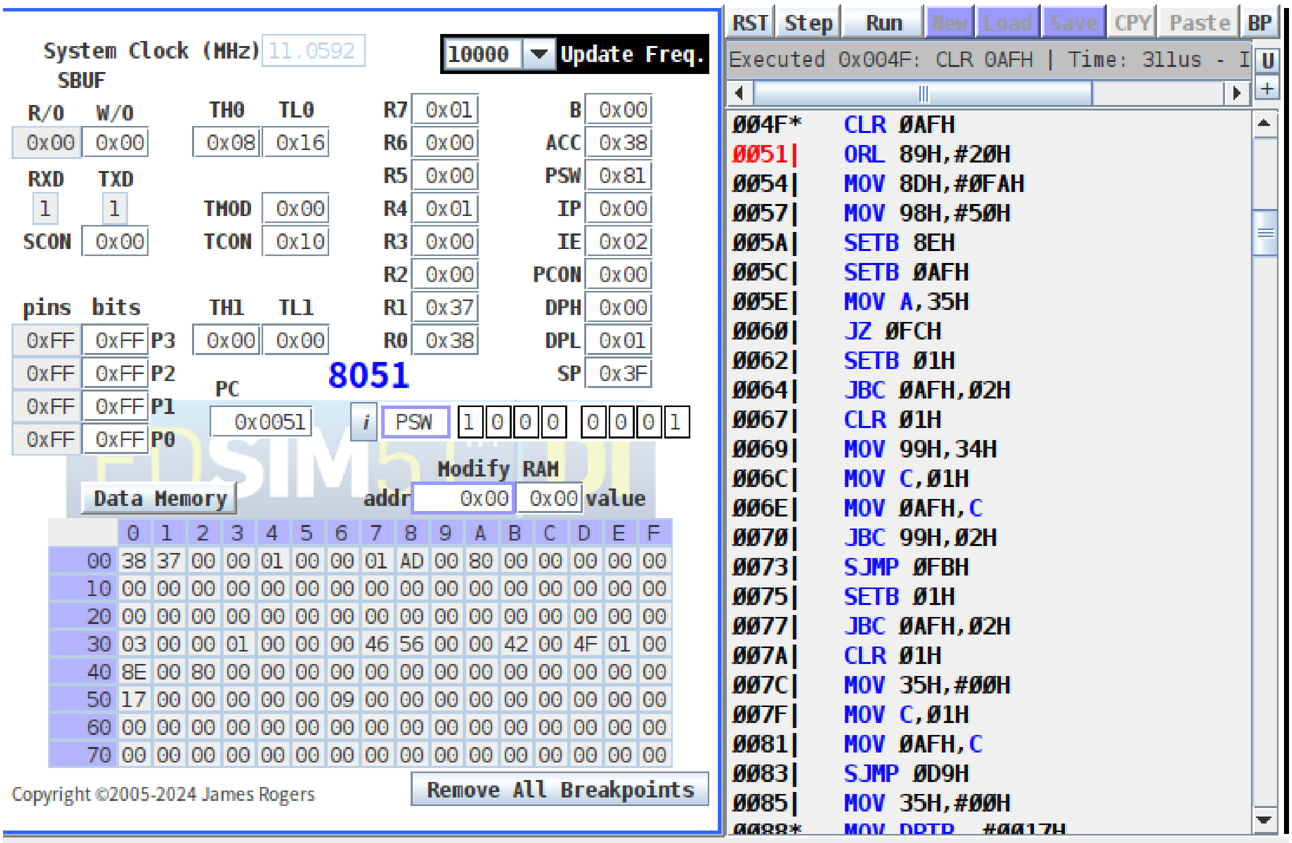
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	37	38	00	00	01	00	00	01	AD	00	80	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	03	01	02	01	00	00	00	46	56	00	00	42	00	4F	01	00
40	60	00	00	00	01	00	80	37	37	00	00	01	00	00	01	00
50	17	00	00	00	00	00	09	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Time: 9ms 32us - Instructions: 5548

```
0017*    MOV 36H,#41H
001A|    MOV A,35H
001C|    JNZ 0FCH
001E|    SETB 00H
0020|    JBC 0AFH,02H
0023|    CLR 00H
0025|    MOV 34H,36H
0028|    MOV 35H,#01H
002B|    MOV C,00H
002D|    MOV 0AFH,C
002F|    MOV A,#5AH
0031|    CJNE A,36H,03H
0034|    SETB C
0035|    SJMP 01H
0037|    CLR C
0038|    MOV 00H,C
003A|    JC 0BH
003C|    MOV R7,36H
003E|    INC R7
003F|    MOV A,R7
0040|    MOV R6,A
0041|    RLC A
0042|    SUBB A,0E0H
0044|    MOV R7,A
0045|    SJMP 04H
```

2. (c)

Consumer is running, because <PSW.4, PSW.3> = 00. Also, this section is initializing the UART.



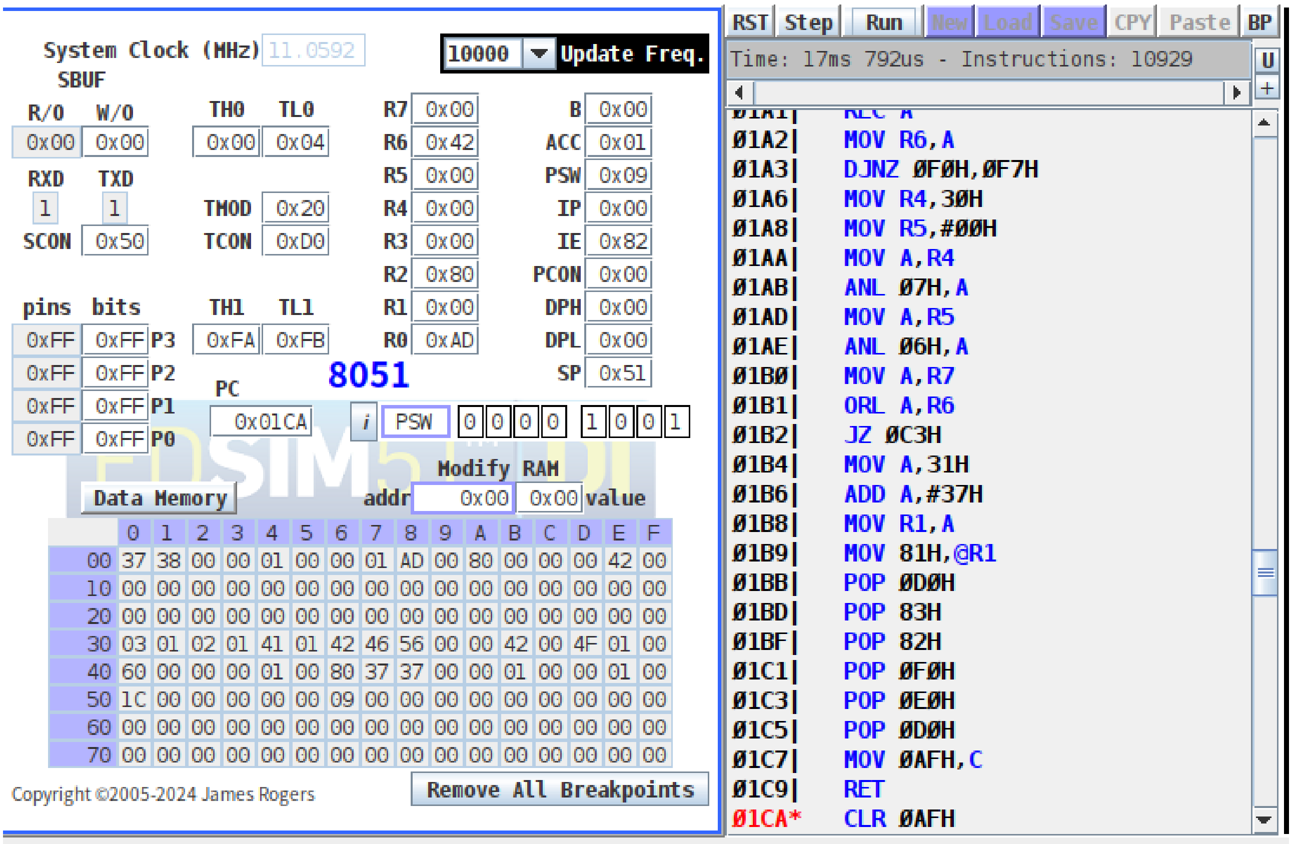
```
Executed 0x004F: CLR 0AFH | Time: 311us - I

System Clock (MHz) 11.0592
SBUF
R/O   W/O     TH0   TL0     R7 0x01     B    0x00
0x00  0x00    0x08  0x16    R6 0x00     ACC  0x38
RXD   TXD                   R5 0x00     PSW  0x81
1     1       TMOD  0x00    R4 0x01     IP   0x00
SCON  0x00    TCON  0x10    R3 0x00     IE   0x02
                            R2 0x00     PCON 0x00
pins  bits    TH1   TL1     R1 0x37     DPH  0x00
0xFF  0xFF P3 0x00  0x00    R0 0x38     DPL  0x01
0xFF  0xFF P2                           SP   0x3F
0xFF  0xFF P1   PC
0xFF  0xFF P0   0x0051   i  PSW 1 0 0 0  0 0 0 1
                          addr 0x00  0x00 value

    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 38 37 00 00 01 00 00 01 AD 00 80 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 03 00 00 01 00 00 00 46 56 00 00 42 00 4F 01 00
40 8E 00 80 00 00 00 00 00 00 00 00 00 00 00 00 00
50 17 00 00 00 00 00 09 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Copyright ©2005-2024 James Rogers
```

```
ØØ4F*  CLR  ØAFH
ØØ51|  ORL  89H,#2ØH
ØØ54|  MOV  8DH,#ØFAH
ØØ57|  MOV  98H,#5ØH
ØØ5A|  SETB 8EH
ØØ5C|  SETB ØAFH
ØØ5E|  MOV  A,35H
ØØ6Ø|  JZ   ØFCH
ØØ62|  SETB Ø1H
ØØ64|  JBC  ØAFH,Ø2H
ØØ67|  CLR  Ø1H
ØØ69|  MOV  99H,34H
ØØ6C|  MOV  C,Ø1H
ØØ6E|  MOV  ØAFH,C
ØØ7Ø|  JBC  99H,Ø2H
ØØ73|  SJMP ØFBH
ØØ75|  SETB Ø1H
ØØ77|  JBC  ØAFH,Ø2H
ØØ7A|  CLR  Ø1H
ØØ7C|  MOV  35H,#ØØH
ØØ7F|  MOV  C,Ø1H
ØØ81|  MOV  ØAFH,C
ØØ83|  SJMP ØD9H
ØØ85|  MOV  35H,#ØØH
ØØ88*  MOV  DPTR,#ØØ17H
```

2. (d)

We can know that the interrupt is triggering on a regular basis by:

(i) Setting the break point at address of myTimer0Handler or the ISR. We can observe that program will jump to myTimer0Handler when (TH0, TL0) = (0x00, 0x04) i.e. timer0 has just been rolled over.



```
Time: 17ms 792us - Instructions: 10929

System Clock (MHz) 11.0592
SBUF
R/O   W/O     TH0   TL0     R7 0x00     B    0x00
0x00  0x00    0x00  0x04    R6 0x42     ACC  0x01
RXD   TXD                   R5 0x00     PSW  0x09
1     1       TMOD  0x20    R4 0x00     IP   0x00
SCON  0x50    TCON  0xD0    R3 0x00     IE   0x82
                            R2 0x80     PCON 0x00
pins  bits    TH1   TL1     R1 0x00     DPH  0x00
0xFF  0xFF P3 0xFA  0xFB    R0 0xAD     DPL  0x00
0xFF  0xFF P2                           SP   0x51
0xFF  0xFF P1   PC
0xFF  0xFF P0   0x01CA   i  PSW 0 0 0 0  1 0 0 1
                          addr 0x00  0x00 value

    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 37 38 00 00 01 00 00 01 AD 00 80 00 00 00 42 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 03 01 02 01 41 01 42 46 56 00 00 42 00 4F 01 00
40 60 00 00 00 01 00 80 37 37 00 00 01 00 00 01 00
50 1C 00 00 00 00 00 09 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Copyright ©2005-2024 James Rogers
```

```
Ø1A2|  MOV  R6,A
Ø1A3|  DJNZ ØFØH,ØF7H
Ø1A6|  MOV  R4,3ØH
Ø1A8|  MOV  R5,#ØØH
Ø1AA|  MOV  A,R4
Ø1AB|  ANL  Ø7H,A
Ø1AD|  MOV  A,R5
Ø1AE|  ANL  Ø6H,A
Ø1BØ|  MOV  A,R7
Ø1B1|  ORL  A,R6
Ø1B2|  JZ   ØC3H
Ø1B4|  MOV  A,31H
Ø1B6|  ADD  A,#37H
Ø1B8|  MOV  R1,A
Ø1B9|  MOV  81H,@R1
Ø1BB|  POP  ØDØH
Ø1BD|  POP  83H
Ø1BF|  POP  82H
Ø1C1|  POP  ØFØH
Ø1C3|  POP  ØEØH
Ø1C5|  POP  ØDØH
Ø1C7|  MOV  ØAFH,C
Ø1C9|  RET
Ø1CA*  CLR  ØAFH
```

(ii)
The fact that the program is switching between producer and consumer.