

## Operating System CP2

許木羽 / 111000177

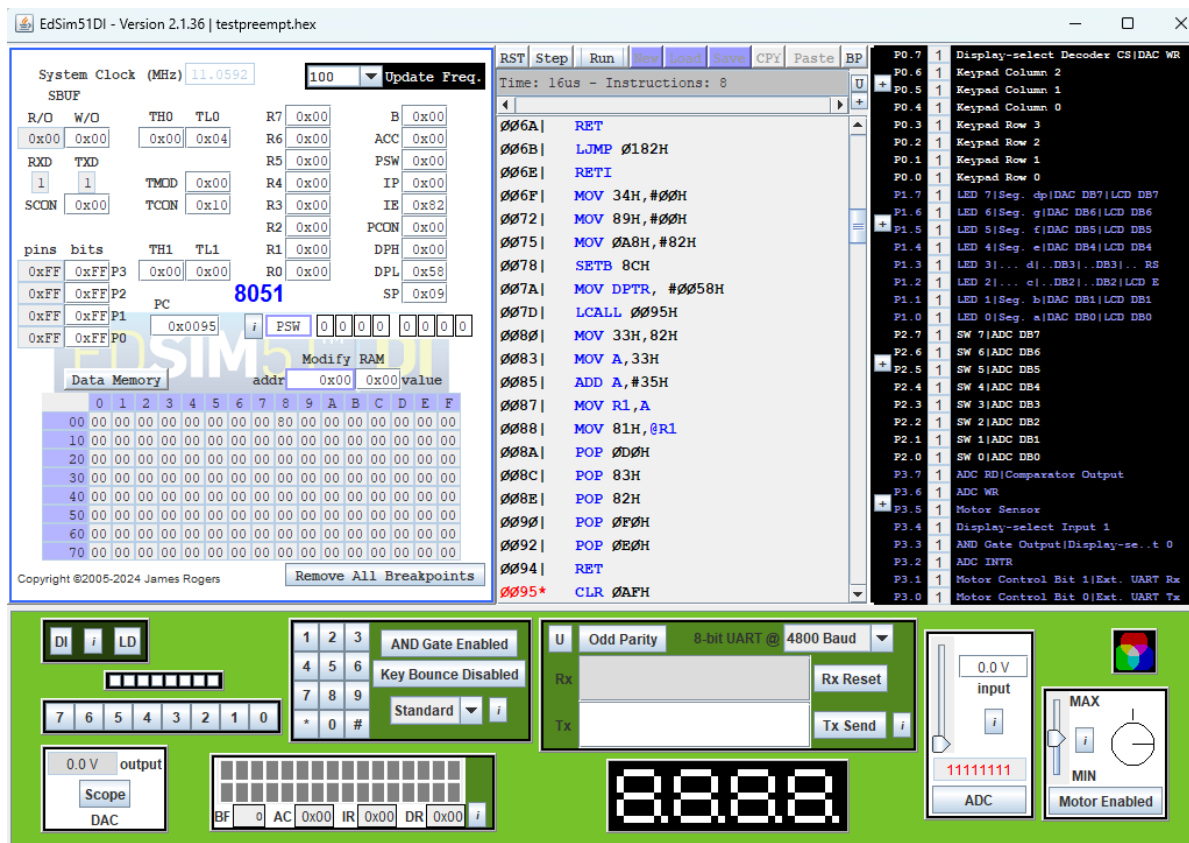
```
PS D:\College\OS\ppc2> mingw32-make clean
del /Q *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym *.asm *.lk
PS D:\College\OS\ppc2> mingw32-make
sdcc -c testpreempt.c
sdcc -c preemptive.c
preemptive.c:93: warning 85: in function ThreadCreate unreferenced function argument : 'fp'
sdcc -o testpreempt.hex testpreempt.rel preemptive.rel
PS D:\College\OS\ppc2> |
```

Similar with CP1, we create the breakpoint for each Consumer function and Producer function. Also, it will output ABCDEFG... in the Rx. I can tell there is an interruption as the output sometimes gets delayed.

In this checkpoint, for every global variable I will disable interrupt by setting variable EA = 0, and then set it back after finish using the function. In myTimer0Handler function, I push every register from r0-r7 and accumulator to the stack. Using round robin scheduling, I find the active thread ID and get into it. After that I restore the previously saved state back into the register. This one to help measure overhead caused by context switching and thread management from incoming interrupt. This function is from myTimer0Handler and will be called regularly

We could know the CreateThread function is called as we put breakpoint from the .map file, which the address located at 0x95

	Value	Global	Global Defined In Module
	-----	-----	-----
C:	00000014	_Producer	testpreempt
C:	0000003A	_Consumer	testpreempt
C:	00000058	_main	testpreempt
C:	00000064	__sdcc_gsinit_startup	testpreempt
C:	00000068	__mcs51_genRAMCLEAR	testpreempt
C:	00000069	__mcs51_genXINIT	testpreempt
C:	0000006A	__mcs51_genXRAMCLEAR	testpreempt
C:	0000006B	_timer0_ISR	testpreempt
C:	0000006F	_Bootstrap	preemptive
C:	00000095	_ThreadCreate	preemptive
C:	00000112	_ThreadYield	preemptive
C:	0000016C	_ThreadExit	preemptive
C:	00000182	_myTimer0Handler	preemptive



Here is the behavior of thread create; the SP will be going to 0x3F and in address code 0xED

```

00E5| ADD A,#3FH
00E7| MOV 81H,A
00E9| PUSH 82H
00EB| PUSH 83H
00ED| MOV A,#00H
00EF| PUSH 0E0H
00F1| PUSH 0E0H
00F3| PUSH 0E0H
00F5| PUSH 0E0H
00F7| MOV A,3AH
00F9| MOV R7,A
00FA| SWAP A
00FB| RR A
00FC| ANL A,#0F8H
00FE| MOV 0D0H,A
0100| PUSH 0D0H
0102| MOV A,3AH
0104| ADD A,#35H
0106| MOV R0,A
0107| MOV @R0,81H
0109| MOV 81H,3CH

```

we can see it clear the accumulator and push accumulator. Here the SP firstly saved to a local variable (not shown in left) however in this code the SP additionally increase and restore back to previous local variables, as my code below leads to that behavior:

```

currentSP = SP;
SP = 0x3F + (0x10 * newThreadID);

asm
PUSH DPL
PUSH DPH
endasm;

asm
MOV A, #0
PUSH ACC
PUSH ACC
PUSH ACC
PUSH ACC
endasm;

PSW = (newThreadID << 3);
asm
PUSH PSW
endasm;

savedStackPointers[newThreadID] = SP;
SP = currentSP;

```

EdSim51DI - Version 2.1.36 | testpreempt.hex

System Clock (MHz): 11.0592 | 100 Update Freq.

SBUP

R/O	W/O	TH0	TL0	R7	B
0x00	0x00	0x05	0x09	0x00	0x00

R6	ACC	R5	PSW	R4	IP
0x00	0x36	0x00	0x08	0x00	0x00

R3	IE	R2	PCON	R1	DPH
0x00	0x82	0x00	0x00	0x00	0x00

R0	DPL	SP
0x36	0x01	0x3F

pins bits TH1 TL1

bits	TH1	TL1
0xFF	0x00	0x00

PC: 0x003A | PSW: 0 0 0 0 1 0 0 0

Modify RAM

Data Memory	addr	0x00	0x00	value												
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00	35	35	00	00	00	00	01	01	36	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	03	46	56	00	00	00	01	00	41	00	00	00
40	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	14	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

Remove All Breakpoints

RST Step Run New Load Save CPY Paste BP

Time: 195us - Instructions: 117

```
003A* ORL 89H, #20H
003D| MOV 8DH, #0FAH
0040| MOV 98H, #50H
0043| SETB 8EH
0045| MOV A, 30H
0047| JZ 0FCH
0049| CLR 0AFH
004B| MOV 99H, 31H
004E| MOV 30H, #00H
0051| SETB 0AFH
0053| JBC 99H, 0EFH
0056| SJMP 0FBH
0058| MOV 30H, #00H
005B| MOV DPTR, #0014H
005E| LCALL 0095H
0061| LJMP 003AH
0064| LJMP 006FH
0067| RET
0068| RET
0069| RET
006A| RET
```

P0.7 1 Display-select Decoder CS/DAC WR  
P0.6 1 Keypad Column 2  
P0.5 1 Keypad Column 1  
P0.4 1 Keypad Column 0  
P0.3 1 Keypad Row 3  
P0.2 1 Keypad Row 2  
P0.1 1 Keypad Row 1  
P0.0 1 Keypad Row 0  
P1.7 1 LED 7(Seg. dp/DAC DB7/LCD DB7  
P1.6 1 LED 6(Seg. g/DAC DB6/LCD DB6  
P1.5 1 LED 5(Seg. f/DAC DB5/LCD DB5  
P1.4 1 LED 4(Seg. e/DAC DB4/LCD DB4  
P1.3 1 LED 3|... d|...DB3|...DB3|... RS  
P1.2 1 LED 2|... c|...DB2|...DB2|... E  
P1.1 1 LED 1(Seg. b/DAC DB1/LCD DB1  
P1.0 1 LED 0(Seg. a/DAC DB0/LCD DB0  
P2.7 1 SW 7(ADC DB7  
P2.6 1 SW 6(ADC DB6  
P2.5 1 SW 5(ADC DB5  
P2.4 1 SW 4(ADC DB4  
P2.3 1 SW 3(ADC DB3  
P2.2 1 SW 2(ADC DB2  
P2.1 1 SW 1(ADC DB1  
P2.0 1 SW 0(ADC DB0  
P3.7 1 ADC RD(Comparator Output  
P3.6 1 ADC WR  
P3.5 1 Motor Sensor  
P3.4 1 Display-select Input 1  
P3.3 1 AND Gate Output(Display-se..t 0  
P3.2 1 ADC INTR  
P3.1 1 Motor Control Bit 1(Ext. UART Rx  
P3.0 1 Motor Control Bit 0(Ext. UART Tx

DI LD

AND Gate Enabled

Key Bounce Disabled

Standard

0.0 V output

U Odd Parity 8-bit UART @ 4800 Baud

Rx Reset

Tx Send

0.0 V input

MAX

MIN

11111111

EdSim51DI - Version 2.1.36 | testpreempt.hex

System Clock (MHz): 11.0592 | 10000 Update Freq.

SBUP

R/O	W/O	TH0	TL0	R7	B
0x00	0x00	0x03	0x15	0x00	0x00

R6	ACC	R5	PSW	R4	IP
0x00	0x00	0x00	0x08	0x00	0x00

R3	IE	R2	PCON	R1	DPH
0x00	0x82	0x00	0x00	0x36	0x00

R0	DPL	SP
0x00	0x00	0x4F

pins bits TH1 TL1

bits	TH1	TL1
0xFF	0xFA	0xFF

PC: 0x0014 | PSW: 0 0 0 0 1 0 0 0

Modify RAM

Data Memory	addr	0x00	0x00	value												
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00	35	35	00	00	00	00	01	01	36	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	01	03	46	56	00	00	00	01	00	41	00	00	00	00
40	45	00	00	01	00	08	00	00	00	00	00	00	00	00	00	00
50	14	00	00	00	00	09	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

Copyright ©2005-2024 James Rogers

Remove All Breakpoints

RST Step Run New Load Save CPY Paste BP

Time: 9ms 28us - Instructions: 5543

```
ORG 0000H
0000| LJMP 0064H
0003| RETI
ORG 000BH
000B| LJMP 006BH
000E| LJMP 0058H
0011| LJMP 000EH
0014* MOV 32H, #41H
0017| MOV A, #01H
0019| CJNE A, 30H, 02H
001C| SJMP 0F9H
001E| CLR 0AFH
0020| MOV 31H, 32H
0023| MOV 30H, #01H
0026| SETB 0AFH
0028| MOV A, #5AH
002A| CJNE A, 32H, 05H
002D| MOV 32H, #41H
0030| SJMP 0E5H
0032| MOV A, 32H
0034| MOV R7, A
```

P0.7 1 Display-select Decoder CS/DAC WR  
P0.6 1 Keypad Column 2  
P0.5 1 Keypad Column 1  
P0.4 1 Keypad Column 0  
P0.3 1 Keypad Row 3  
P0.2 1 Keypad Row 2  
P0.1 1 Keypad Row 1  
P0.0 1 Keypad Row 0  
P1.7 1 LED 7(Seg. dp/DAC DB7/LCD DB7  
P1.6 1 LED 6(Seg. g/DAC DB6/LCD DB6  
P1.5 1 LED 5(Seg. f/DAC DB5/LCD DB5  
P1.4 1 LED 4(Seg. e/DAC DB4/LCD DB4  
P1.3 1 LED 3|... d|...DB3|...DB3|... RS  
P1.2 1 LED 2|... c|...DB2|...DB2|... E  
P1.1 1 LED 1(Seg. b/DAC DB1/LCD DB1  
P1.0 1 LED 0(Seg. a/DAC DB0/LCD DB0  
P2.7 1 SW 7(ADC DB7  
P2.6 1 SW 6(ADC DB6  
P2.5 1 SW 5(ADC DB5  
P2.4 1 SW 4(ADC DB4  
P2.3 1 SW 3(ADC DB3  
P2.2 1 SW 2(ADC DB2  
P2.1 1 SW 1(ADC DB1  
P2.0 1 SW 0(ADC DB0  
P3.7 1 ADC RD(Comparator Output  
P3.6 1 ADC WR  
P3.5 1 Motor Sensor  
P3.4 1 Display-select Input 1  
P3.3 1 AND Gate Output(Display-se..t 0  
P3.2 1 ADC INTR  
P3.1 1 Motor Control Bit 1(Ext. UART Rx  
P3.0 1 Motor Control Bit 0(Ext. UART Tx

DI LD

AND Gate Enabled

Key Bounce Disabled

Standard

0.0 V output

U Odd Parity 8-bit UART @ 4800 Baud

Rx Reset

Tx Send

0.0 V input

MAX

MIN

11111111

From the previous page, above is the Consumer Thread and Below is the Consumer Thread. When we are working on Producer Thread, the SP will change to 0x4F, but Consumer Thread is at 0x3F, and this is reasonable since they shared the different thread.

