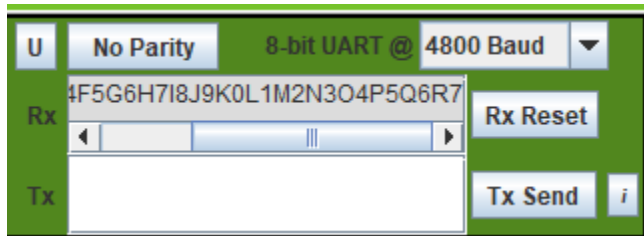


Operating System CP4

許木羽 / 111000177



The semaphore like CP3, however there is an extra semaphore variable to take turn for the producer. Here as one producer output ABCDEFG..., and the other producer output 01234..., my fairness system is to output everything in turn, so the expected output will be:

A0B1C2D3E4F5...

There will be new 2 semaphore variable, let say variable A and B. For producer 1, it needs to wait the signal from A and then after finishing sending signal B. Producer 2 will wait for the signal B and then after finish, send the signal A. Therefore, the turn will be correct.

	Value	Global	Global Defined In Module
C:	00000014	_Producer1	test3preempt
C:	00000065	_Producer2	test3preempt
C:	000000B6	_Consumer	test3preempt
C:	000000F7	_main	test3preempt
C:	0000011B	__sdcc_gsinit_startup	test3preempt
C:	0000011F	__mcs51_genRAMCLEAR	test3preempt
C:	00000120	__mcs51_genXINIT	test3preempt
C:	00000121	__mcs51_genXRAMCLEAR	test3preempt
C:	00000122	_timer0_ISR	test3preempt
C:	00000126	_Bootstrap	preemptive
C:	0000014C	_ThreadCreate	preemptive
C:	000001C9	_ThreadYield	preemptive
C:	00000223	_ThreadExit	preemptive
C:	00000239	_myTimer0Handler	preemptive

System Clock (MHz)11.059210000Update Freq.

SBUF

R/OW/O

0x000x00

TH0TL0

0x040x03

R7

0x02

B

0x00

R6

0x03

ACC

0x03

R5

0x00

PSW

0x08

R4

0x00

IP

0x00

R3

0x00

IE

0x82

R2

0x00

PCON

0x00

R1

0x01

DPH

0x00

R0

0x36

DPL

0x00

SP

0x4F

RXDTXD

11

TMOD

0x20

SCON

0x50

TCON

0xD0

pinsbits

TH1TL1

0xFA0xFD

PC

8051

0x0027

PSW

000010000

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	01	00	01	36	01	00	00	00	00	03	02
10	00	36	00	00	00	00	00	01	00	00	00	00	00	00	00	00
20	01	00	03	00	00	00	00	53	52	37	00	00	00	00	00	00
30	00	41	38	01	07	46	56	66	00	00	02	00	41	00	00	00
40	C1	00	00	00	02	00	10	00	00	00	00	01	00	00	00	00
50	14	00	00	00	00	00	09	00	00	00	03	52	00	00	00	00
60	65	00	00	00	00	00	11	00	00	00	00	37	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Remove All Breakpoints

RSTStepRunNewLoadSaveCPYPasteBP

Executed 0x0024: JB 0E7H,0F9H | Time: 9ms 4

0011|LJMP 000EH

0014|MOV 31H,#41H

0017|MOV A,25H

0019*JZ 0FCH

001B|JB 0E7H,0F9H

001E|DEC 25H

0020|MOV A,22H

0022|JZ 0FCH

0024|JB 0E7H,0F9H

0027|DEC 22H

0029|MOV A,20H

002B|JZ 0FCH

002D|JB 0E7H,0F9H

0030|DEC 20H

0032|CLR 0AFH

0034|MOV A,24H

0036|ADD A,#27H

0038|MOV R0,A

0039|MOV @R0,31H

003B|SETB 0AFH

003D|INC 20H

System Clock (MHz)11.059210000Update Freq.

SBUF

R/OW/O

0x000x00

TH0TL0

0x050x10

R7

0x41

B

0x00

R6

0x03

ACC

0x00

R5

0x00

PSW

0x08

R4

0x00

IP

0x00

R3

0x00

IE

0x82

R2

0x00

PCON

0x00

R1

0x01

DPH

0x00

R0

0x27

DPL

0x00

SP

0x4F

RXDTXD

11

TMOD

0x20

SCON

0x50

TCON

0xD0

pinsbits

TH1TL1

0xFA0xFA

PC

8051

0x0017

PSW

000010000

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	01	00	01	27	01	00	00	00	00	03	41
10	00	36	00	00	00	00	00	01	00	00	00	00	00	00	00	00
20	01	01	02	00	01	00	01	41	52	37	00	00	00	00	00	00
30	00	42	38	01	07	46	56	66	00	00	02	00	41	00	00	00
40	C1	00	00	00	02	00	10	00	00	00	00	01	00	00	00	00
50	14	00	00	00	00	00	09	00	00	00	03	52	00	00	00	00
60	65	00	00	00	00	00	11	00	00	00	00	37	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Remove All Breakpoints

RSTStepRunNewLoadSaveCPYPasteBP

Executed 0x0019: JZ 0FCH | Time: 9ms 92us

ORG 0000H

0000|LJMP 011BH

0003|RETI

ORG 000BH

000B|LJMP 0122H

000E|LJMP 00F7H

0011|LJMP 000EH

0014|MOV 31H,#41H

0017|MOV A,25H

0019*JZ 0FCH

001B|JB 0E7H,0F9H

001E|DEC 25H

0020|MOV A,22H

0022|JZ 0FCH

0024|JB 0E7H,0F9H

0027|DEC 22H

0029|MOV A,20H

002B|JZ 0FCH

002D|JB 0E7H,0F9H

0030|DEC 20H

0032|CLR 0AFH

Above image is producer 1 first iteration and below is the second iteration. From the image, we can see that the first will not stall while the second will stall. This is because the producer 1 already did the output and haven't wait for the producer 2 to output. After some stalling it will go do the producer 2 first as below image

The screenshot displays the Proteus IDE interface for an 8051 microcontroller. The left pane shows the configuration for the 8051, including the System Clock (11.0592 MHz), SBUF, and various registers. The right pane shows the assembly code for the program.

8051 Configuration:

- System Clock (MHz): 11.0592
- Update Freq: 10000
- Registers: R7 (0x01), R6 (0x00), R5 (0x00), R4 (0x00), R3 (0x00), R2 (0x00), R1 (0x36), R0 (0x00), ACC (0x01), PSW (0x11), IP (0x00), IE (0x82), PCON (0x00), DPH (0x00), DPL (0x00), SP (0x5F)
- TH0 (0x04), TL0 (0x05), TMOD (0x20), TCON (0xD0), TH1 (0xFA), TL1 (0xFB), PC (0x0071)
- PSW: 0 0 0 1 0 0 0 1

Assembly Code:

```

0053 | MOV A, #5AH
0055 | CJNE A, 31H, 05H
0058 | MOV 31H, #41H
005B | SJMP 0BAH
005D | MOV A, 31H
005F | MOV R7, A
0060 | INC A
0061 | MOV 31H, A
0063 | SJMP 0B2H
0065 | MOV 32H, #30H
0068* | MOV A, 26H
006A | JZ 0FCH
006C | JB 0E7H, 0F9H
006F | DEC 26H
0071 | MOV A, 22H
0073 | JZ 0FCH
0075 | JB 0E7H, 0F9H
0078 | DEC 22H
007A | MOV A, 20H
007C | JZ 0FCH

```

Below is the consumer

System Clock (MHz)11.059210000Update Freq.

SBUF

R/O W/O

0x00 0x00

TH0 TL0

0x09 0x0C

R7

0x01

B

0x00

R6

0x00

ACC

0x00

R5

0x00

PSW

0x10

R4

0x00

IP

0x00

R3

0x00

IE

0x82

R2

0x00

PCON

0x00

R1

0x36

DPH

0x00

R0

0x37

DPL

0x02

SP

0x3F

RXD TXD

1 1

TMOD

0x20

TCON

0x50

pins bits

TH1 TL1

0xFA 0x1E

PC

0x00C1

PSW

0 0 0 1 0 0 0 0

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	01	01	01	36	01	00	00	00	00	03	02
10	37	36	00	00	00	00	00	01	00	00	00	00	00	00	00	00
20	01	00	03	00	00	01	00	41	52	37	00	00	00	00	00	00
30	00	42	30	00	07	46	56	66	00	00	02	00	41	00	00	00
40	18	01	00	00	00	00	00	00	00	00	00	01	00	00	00	00
50	14	00	00	00	00	00	09	00	00	00	03	41	00	00	00	00
60	65	00	00	00	00	00	11	00	00	00	00	37	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Remove All Breakpoints

Executed 0x00C3: JZ 0FCH | Time: 337us - In

MOV R7,A

INC A

MOV 32H,A

SJMP 0B2H

ORL 89H,#20H

MOV 8DH,#0FAH

MOV 98H,#50H

SETB 8EH

MOV A,21H

JZ 0FCH

JB 0E7H,0F9H

DEC 21H

MOV A,20H

JZ 0FCH

JB 0E7H,0F9H

DEC 20H

CLR 0AFH

MOV A,23H

ADD A,#27H

MOV R1,A

MOV 99H,@R1

System Clock (MHz)11.059210000Update Freq.

SBUF

R/O W/O

0x00 0x00

TH0 TL0

0x04 0x0E

R7

0x30

B

0x00

R6

0x00

ACC

0x02

R5

0x00

PSW

0x35

R4

0x00

IP

0x00

R3

0x00

IE

0x82

R2

0x00

PCON

0x00

R1

0x38

DPH

0x00

R0

0x00

DPL

0x00

SP

0xF9

RXD TXD

1 1

TMOD

0x20

TCON

0xD0

pins bits

TH1 TL1

0xFA 0xFA

PC

0x00CA

PSW

0 0 1 1 0 1 0 1

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	01	00	01	00	37	00	00	00	00	03	41
10	00	38	00	00	00	00	00	30	00	00	00	00	00	00	00	00
20	01	01	01	00	02	01	00	41	30	37	00	00	00	00	00	00
30	00	42	31	03	07	46	56	66	00	00	02	00	41	00	00	00
40	C1	00	00	00	02	00	10	00	00	00	00	01	00	00	00	00
50	17	00	00	00	00	00	08	00	00	00	03	41	00	00	00	00
60	68	00	00	00	00	00	10	00	00	00	00	30	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Remove All Breakpoints

Time: 26ms 833us - Instructions: 16494

ORL 89H,#20H

MOV 8DH,#0FAH

MOV 98H,#50H

SETB 8EH

MOV A,21H

JZ 0FCH

JB 0E7H,0F9H

DEC 21H

MOV A,20H

JZ 0FCH

JB 0E7H,0F9H

DEC 20H

CLR 0AFH

MOV A,23H

ADD A,#27H

MOV R1,A

MOV 99H,@R1

SETB 0AFH

INC 20H

INC 22H

MOV A,#02H

First image is the consumer get stalled, and below is the consumer succeed from the full signal.

- Makefile
- preemptive.asm
- preemptive.c
- preemptive.h
- preemptive.lst
- preemptive.rel
- preemptive.rst
- preemptive.sym
- test3preempt.asm
- test3preempt.c
- test3preempt.hex
- test3preempt.lk
- test3preempt.lst
- test3preempt.map
- test3preempt.mem
- test3preempt.rel
- test3preempt.rst
- test3preempt.sym