

Assignment Report 3

1. Solution

1.1 Initiate the stack pointer

This is simple, just a LEA instruction can do it, and use .BLKW to apply some stack space.

```
1  .ORIG x3000
2  LEA R6,STACK
3  ; initialize the stack pointer
4
```

1.2 Set up the keyboard interrupt

This is also simple, set the address x0180 (x0100+x0080, which is the keyboard interrupt vector) as x2000, which is the starting address of the interrupt service routine.

```
4
5  LD R2,START
6  STI R2,KB
7  ; set up the keyboard interrupt
8  ; vector table entry (set x0180's
9  ; value as x2000, which is the
10 ; starting address of the interrupt
11 ; routine)
12
```

1.3 Enable keyboard interrupts

This needs to set KBSR[14] as 1, which is also simple, use De Morgan's law can do it.

```
12
13  LDI R0,KBSR
14  LD R1,MASK
15  NOT R0,R0
16  NOT R1,R1
17  AND R0,R0,R1
18  NOT R0,R0
19  STI R0,KBSR
20 ; enable keyboard interrupts,
21 ; which is set KBSR[14] as 1
22
```

1.4 Actual user program to print the checkerboard

I set an infinite loop to do it. The loop contains two sub-programs, representing two different lines. As for which pattern, I use a R5 (0 or 1) as a flag to judge, if R5 is 1 then it is the first pattern, or it is the second pattern, and R5 would be changed in the interrupt service routine.

```
22
23     AND R5,R5,#0
24     ADD R5,R5,#1
25 LOOP JSR LINE1
26     JSR LINE2
27     BRnzp LOOP
28     TRAP x25 ;actually it's useless
29 ;
```

```
29 ;
30 LINE1 ST R0,SR0
31     ST R1,SR1
32     ST R2,SR2
33     ST R3,SR3
34     ST R4,SR3
35     ST R5,SR5
36     ST R6,SR6
37     ST R7,SR7
38     ADD R5,R5,#0
39     BRp PT1_1
40     BRz PT1_2
41 PT1_1 LEA R0,P1
42     BRnzp NEXT1
43 PT1_2 LEA R0,P2
44 NEXT1 AND R1,R1,#0
45     ADD R1,R1,#8
46 L1 TRAP x22
47     ADD R1,R1,xFFFF
48     BRz DONE
49     BRnzp L1
50 DONE LDI R3,DSR
51     BRzp DONE
52     LD R2,NEWLINE
53     STI R2,DDR
54     LD R0,SR0
55     LD R1,SR1
56     LD R2,SR2
57     LD R3,SR3
58     LD R4,SR4
59     LD R5,SR5
60     LD R6,SR6
61     LD R7,SR7
62     BRnzp DELAY
63     RET
64 ;
```

```
65 LINE2 ST R0,SR0
66     ST R1,SR1
67     ST R2,SR2
68     ST R3,SR3
69     ST R4,SR3
70     ST R5,SR5
71     ST R6,SR6
72     ST R7,SR7
73     ADD R5,R5,#0
74     BRp PA2_1
75     BRz PA2_2
76 PA2_1 LEA R0,P1
77     BRnzp NEXT2
78 PA2_2 LEA R0,P2
79 NEXT2 LD R2,SPACE
80 WRITE LDI R3,DSR
81     BRzp WRITE
82     STI R2,DDR
83     STI R2,DDR
84     STI R2,DDR
85 ; write 3 spaces
86     AND R1,R1,#0
87     ADD R1,R1,#7
88 ; count down
89 L2 TRAP x22
90     ADD R1,R1,xFFFF
91     BRz DONE2
92     BRnzp L2
93 DONE2 LDI R3,DSR
94     BRzp DONE
95     LD R2,NEWLINE
96     STI R2,DDR
97     LD R0,SR0
98     LD R1,SR1
99     LD R2,SR2
100    LD R3,SR3
101    LD R4,SR4
102    LD R5,SR5
103    LD R6,SR6
104    LD R7,SR7
105    BRnzp DELAY
106    RET
```

```

107 ;
108 DELAY ST R1,SR1
109 LD R1,COUNT
110 REP ADD R1,R1,#-1
111 BRp REP
112 LD R1,SR1
113 RET
114 ;
115 P1 .STRINGZ "** "
116 P2 .STRINGZ "## "
117 COUNT .FILL x4FFF
118 NEWLINE .FILL x000A
119 SPACE .FILL #32
120 MASK .FILL x4000
121 DSR .FILL xFE04
122 DDR .FILL xFE06
123 KBSR .FILL xFE00
124 KBDR .FILL xFE02
125 START .FILL x2000
126 KB .FILL x0180
127 SR0 .BLKW 1
128 SR1 .BLKW 1
129 SR2 .BLKW 1
130 SR3 .BLKW 1
131 SR4 .BLKW 1
132 SR5 .BLKW 1
133 SR6 .BLKW 1
134 SR7 .BLKW 1
135 STACK .BLKW 10
136 ; start of actual user program to print the checkerboard
137 .END

```

1.5 Interrupt service routine

This is simple, just write the data that is already in KBDR and repeat 10 times, and change the value of R5, then RTI.

<pre> 1 .ORIG x2000 2 INTERRUPT ST R0,SR0 3 ST R1,SR1 4 ST R2,SR2 5 ST R3,SR3 6 ST R4,SR4 7 ST R5,SR5 8 ST R6,SR6 9 ST R7,SR7 10 LDI R0,KBDR 11 LD R1,COUNT 12 WRITE LDI R3,DSR 13 BRzp WRITE 14 L STI R0,DDR 15 ADD R1,R1,xFFFF 16 BRz DONE 17 BRnzp L 18 DONE LDI R3,DSR 19 BRzp DONE 20 LD R0,NEWLINE 21 STI R0,DDR 22 LD R0,SR0 23 LD R1,SR1 24 LD R2,SR2 25 LD R3,SR3 26 LD R4,SR4 27 LD R5,SR5 28 LD R6,SR6 29 LD R7,SR7 30 ADD R5,R5,#0 31 BRz CHANGE1 32 BRp CHANGE2 33 CHANGE1 ADD R5,R5,#1 34 BRnzp END 35 CHANGE2 ADD R5,R5,xFFFF 36 END RTI </pre>	<pre> 37 38 NEWLINE .FILL x000A 39 KBDR .FILL xFE02 40 DSR .FILL xFE04 41 DDR .FILL xFE06 42 COUNT .FILL #10 43 SR0 .BLKW 1 44 SR1 .BLKW 1 45 SR2 .BLKW 1 46 SR3 .BLKW 1 47 SR4 .BLKW 1 48 SR5 .BLKW 1 49 SR6 .BLKW 1 50 SR7 .BLKW 1 51 .END </pre>
--	---

2. Source Code

user_program:

.ORIG x3000

LEA R6,STACK

; initialize the stack pointer

LD R2,START

STI R2,KB

; set up the keyboard interrupt

; vector table entry (set x0180's

; value as x2000, which is the

; starting address of the interrupt

; routine)

LDI R0,KBSR

LD R1,MASK

NOT R0,R0

NOT R1,R1

AND R0,R0,R1

NOT R0,R0

STI R0,KBSR

; enable keyboard interrupts,

; which is set KBSR[14] as 1

AND R5,R5,#0

ADD R5,R5,#1

LOOP JSR LINE1

JSR LINE2

BRnzp LOOP

TRAP x25 ;actually it's useless

;

LINE1 ST R0,SR0

ST R1,SR1

ST R2,SR2

ST R3,SR3

ST R4,SR3

ST R5,SR5

ST R6,SR6

ST R7,SR7

ADD R5,R5,#0

BRp PT1_1

BRz PT1_2

PT1_1 LEA R0,P1

BRnzp NEXT1	;
PT1_2 LEA R0,P2	LINE2 ST R0,SR0
NEXT1 AND R1,R1,#0	ST R1,SR1
ADD R1,R1,#8	ST R2,SR2
L1 TRAP x22	ST R3,SR3
ADD R1,R1,xFFFF	ST R4,SR3
BRz DONE	ST R5,SR5
BRnzp L1	ST R6,SR6
DONE LDI R3,DSR	ST R7,SR7
BRzp DONE	ADD R5,R5,#0
LD R2,NEWLINE	BRp PA2_1
STI R2,DDR	BRz PA2_2
LD R0,SR0	PA2_1 LEA R0,P1
LD R1,SR1	BRnzp NEXT2
LD R2,SR2	PA2_2 LEA R0,P2
LD R3,SR3	NEXT2 LD R2,SPACE
LD R4,SR4	WRITE LDI R3,DSR
LD R5,SR5	BRzp WRITE
LD R6,SR6	STI R2,DDR
LD R7,SR7	STI R2,DDR
BRnzp DELAY	STI R2,DDR
RET	; write 3 spaces

AND R1,R1,#0	DELAY ST R1,SR1
ADD R1,R1,#7	LD R1,COUNT
; count down	REP ADD R1,R1,#-1
L2 TRAP x22	BRp REP
ADD R1,R1,xFFFF	LD R1,SR1
BRz DONE2	RET
BRnzp L2	;
DONE2 LDI R3,DSR	P1 .STRINGZ "** "
BRzp DONE	P2 .STRINGZ "## "
LD R2,NEWLINE	COUNT .FILL x4FFF
STI R2,DDR	NEWLINE .FILL x000A
LD R0,SR0	SPACE .FILL #32
LD R1,SR1	MASK .FILL x4000
LD R2,SR2	DSR .FILL xFE04
LD R3,SR3	DDR .FILL xFE06
LD R4,SR4	KBSR .FILL xFE00
LD R5,SR5	KBDR .FILL xFE02
LD R6,SR6	START .FILL x2000
LD R7,SR7	KB .FILL x0180
BRnzp DELAY	SR0 .BLKW 1
RET	SR1 .BLKW 1
;	SR2 .BLKW 1

SR3 .BLKW 1	WRITE LDI R3,DSR
SR4 .BLKW 1	BRzp WRITE
SR5 .BLKW 1	L STI R0,DDR
SR6 .BLKW 1	ADD R1,R1,xFFFF
SR7 .BLKW 1	BRz DONE
STACK .BLKW 10	BRnzp L
; start of actual user program to print	DONE LDI R3,DSR
the checkerboard	BRzp DONE
.END	LD R0,NEWLINE
interrupt_service_routine:	STI R0,DDR
.ORIG x2000	LD R0,SR0
INTERRUPTED ST R0,SR0	LD R1,SR1
ST R1,SR1	LD R2,SR2
ST R2,SR2	LD R3,SR3
ST R3,SR3	LD R4,SR4
ST R4,SR3	LD R5,SR5
ST R5,SR5	LD R6,SR6
ST R6,SR6	LD R7,SR7
ST R7,SR7	ADD R5,R5,#0
LDI R0,KBDR	BRz CHANGE1
LD R1,COUNT	BRp CHANGE2
	CHANGE1 ADD R5,R5,#1

```
BRnzp END  
CHANGE2 ADD R5,R5,xFFFF  
END RTI
```

```
NEWLINE .FILL x000A
```

```
KBDR .FILL xFE02
```

```
DSR .FILL xFE04
```

```
DDR .FILL xFE06
```

```
COUNT .FILL #10
```

```
SR0 .BLKW 1
```

```
SR1 .BLKW 1
```

```
SR2 .BLKW 1
```

```
SR3 .BLKW 1
```

```
SR4 .BLKW 1
```

```
SR5 .BLKW 1
```

```
SR6 .BLKW 1
```

```
SR7 .BLKW 1
```

```
.END
```