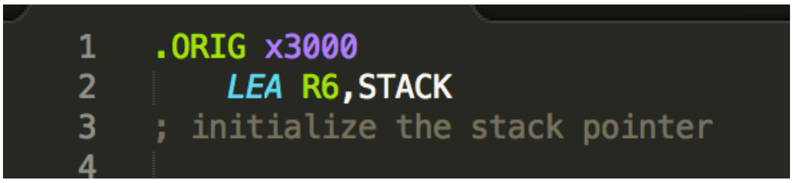
Assignment Report 3

1. Solution

1.1 Initiate the stack pointer

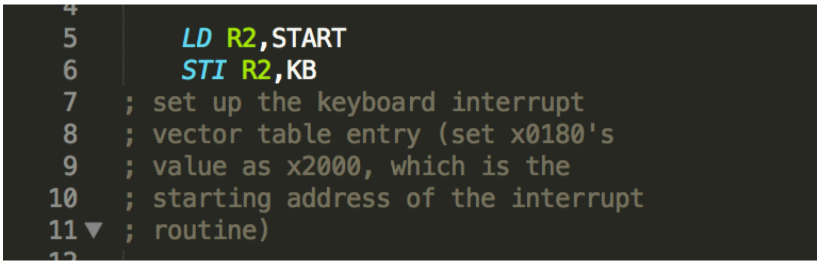
This is simple, just a LEA instruction can do it, and use .BLKW to apply some stack space.



```
.ORIG x3000
    LEA R6,STACK
; initialize the stack pointer
```

1.2 Set up the keyboard interrupt

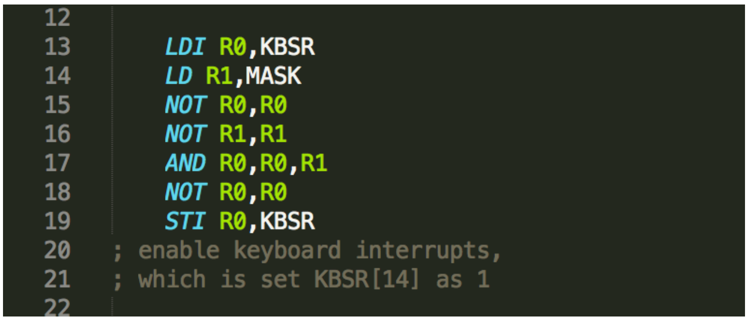
This is also simple, set the address x0180 (x0100+x0080, which is the keyboard interrupt vector) as x2000, which is the starting address of the interrupt service routine.



```
    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)
```

1.3 Enable keyboard interrupts

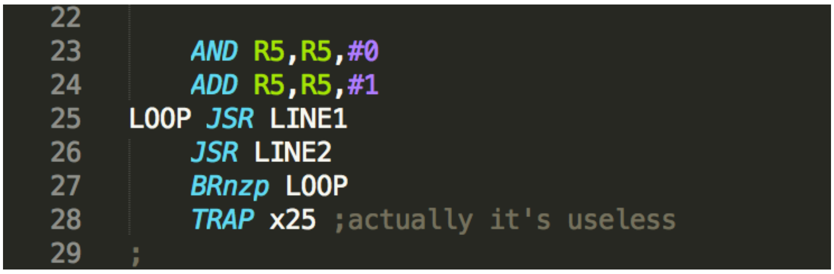
This needs to set KBSR[14] as 1, which is also simple, use De Morgan’s law can do it.



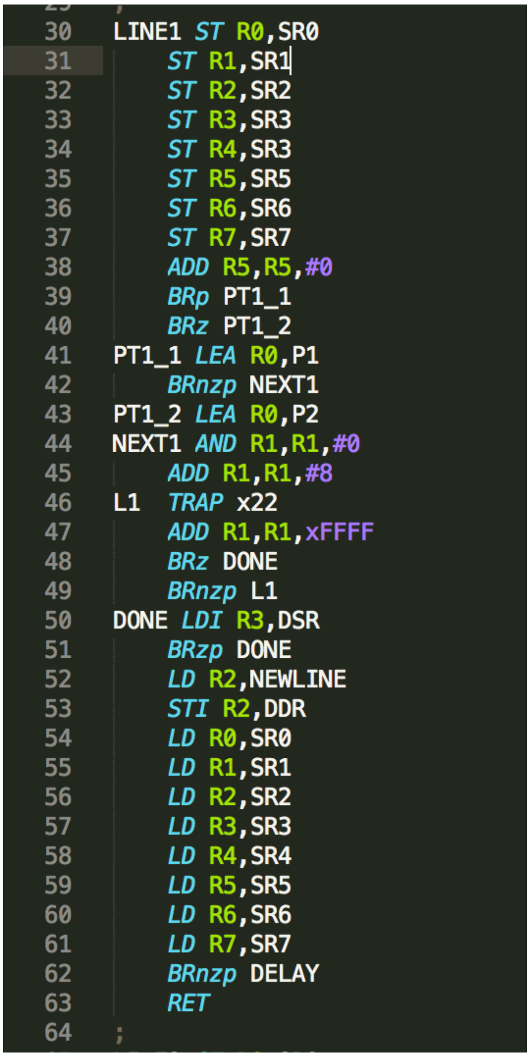
```
    LDI R0,KBSR
    LD R1,MASK
    NOT R0,R0
    NOT R1,R1
    AND R0,R0,R1
    NOT R0,R0
    STI R0,KBSR
; enable keyboard interrupts,
; which is set KBSR[14] as 1
```

1.4 Actual user program to print the checkerboard

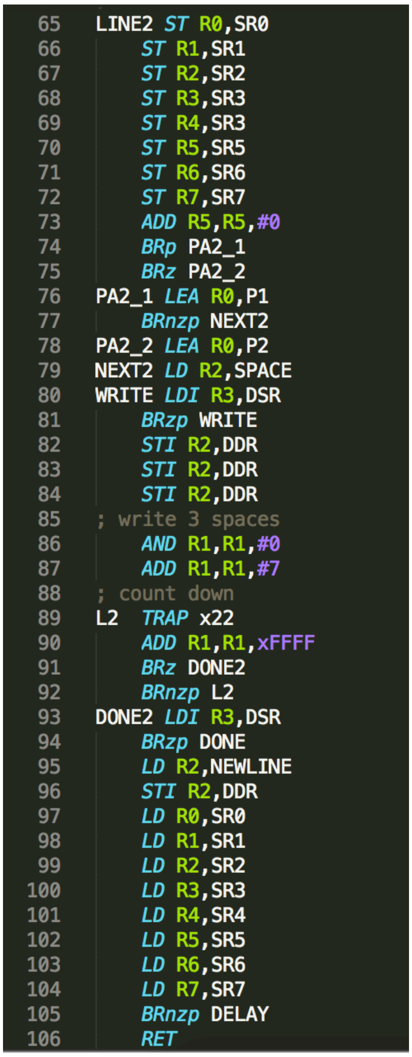
I set an infinite loop to do it. The loop contains two sub-programs, representing two different lines. As for which pattern, I use a R5 (0 or 1) as a flag to judge, if R5 is 1 then it is the first pattern, or it is the second pattern, and R5 would be changed in the interrupt service routine.



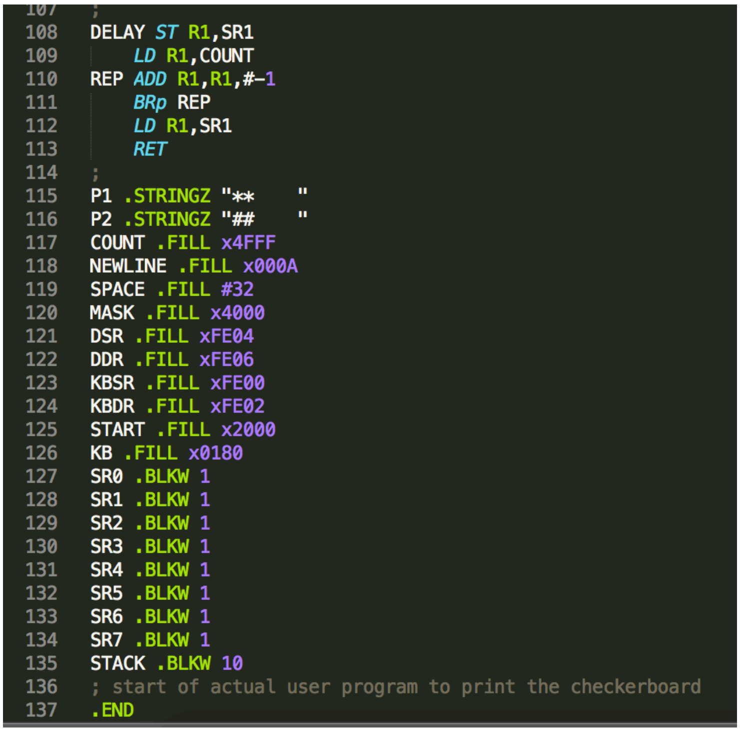
```

        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
NEXT1 AND R1,R1,#0
      ADD R1,R1,#8
L1    TRAP x22
      ADD R1,R1,xFFFF
      BRz DONE
      BRnzp L1
DONE LDI R3,DSR
      BRzp DONE
      LD R2,NEWLINE
      STI R2,DDR
      LD R0,SR0
      LD R1,SR1
      LD R2,SR2
      LD R3,SR3
      LD R4,SR4
      LD R5,SR5
      LD R6,SR6
      LD R7,SR7
      BRnzp DELAY
      RET
;
```



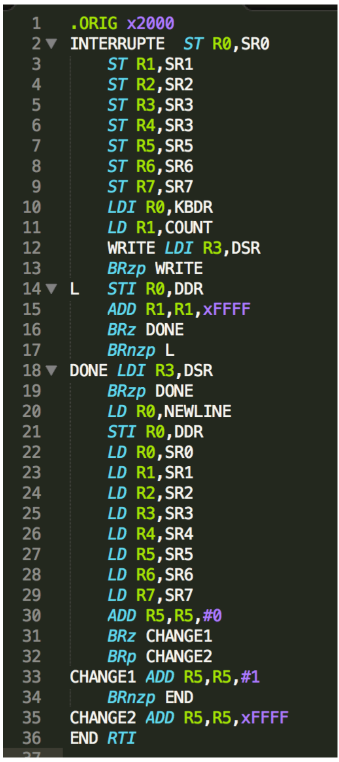
```
LINE2 ST R0,SR0
      ST R1,SR1
      ST R2,SR2
      ST R3,SR3
      ST R4,SR3
      ST R5,SR5
      ST R6,SR6
      ST R7,SR7
      ADD R5,R5,#0
      BRp PA2_1
      BRz PA2_2
PA2_1 LEA R0,P1
      BRnzp NEXT2
PA2_2 LEA R0,P2
NEXT2 LD R2,SPACE
WRITE LDI R3,DSR
      BRzp WRITE
      STI R2,DDR
      STI R2,DDR
      STI R2,DDR
; write 3 spaces
      AND R1,R1,#0
      ADD R1,R1,#7
; count down
L2    TRAP x22
      ADD R1,R1,xFFFF
      BRz DONE2
      BRnzp L2
DONE2 LDI R3,DSR
      BRzp DONE
      LD R2,NEWLINE
      STI R2,DDR
      LD R0,SR0
      LD R1,SR1
      LD R2,SR2
      LD R3,SR3
      LD R4,SR4
      LD R5,SR5
      LD R6,SR6
      LD R7,SR7
      BRnzp DELAY
      RET
```



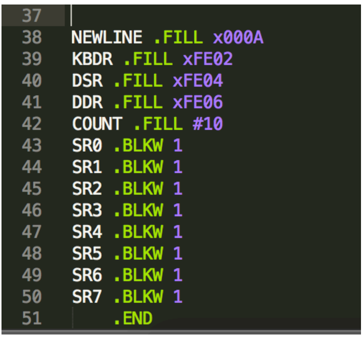
```
;
DELAY ST R1,SR1
    LD R1,COUNT
REP ADD R1,R1,#-1
    BRp REP
    LD R1,SR1
    RET
;
P1 .STRINGZ "**    "
P2 .STRINGZ "##    "
COUNT .FILL x4FFF
NEWLINE .FILL x000A
SPACE .FILL #32
MASK .FILL x4000
DSR .FILL xFE04
DDR .FILL xFE06
KBSR .FILL xFE00
KBDR .FILL xFE02
START .FILL x2000
KB .FILL x0180
SR0 .BLKW 1
SR1 .BLKW 1
SR2 .BLKW 1
SR3 .BLKW 1
SR4 .BLKW 1
SR5 .BLKW 1
SR6 .BLKW 1
SR7 .BLKW 1
STACK .BLKW 10
; start of actual user program to print the checkerboard
.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.



```
.ORIG x2000
INTERRUPTE  ST R0,SR0
    ST R1,SR1
    ST R2,SR2
    ST R3,SR3
    ST R4,SR3
    ST R5,SR5
    ST R6,SR6
    ST R7,SR7
    LDI R0,KBDR
    LD R1,COUNT
    WRITE LDI R3,DSR
    BRzp WRITE
L   STI R0,DDR
    ADD R1,R1,xFFFF
    BRz DONE
    BRnzp L
DONE LDI R3,DSR
    BRzp DONE
    LD R0,NEWLINE
    STI R0,DDR
    LD R0,SR0
    LD R1,SR1
    LD R2,SR2
    LD R3,SR3
    LD R4,SR4
    LD R5,SR5
    LD R6,SR6
    LD R7,SR7
    ADD R5,R5,#0
    BRz CHANGE1
    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
```

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
NEXT1 AND R1,R1,#0
      ADD R1,R1,#8
L1    TRAP x22
      ADD R1,R1,xFFFF
      BRz DONE
      BRnzp L1
DONE LDI R3,DSR
      BRzp DONE
      LD R2,NEWLINE
      STI R2,DDR
      LD R0,SR0
      LD R1,SR1
      LD R2,SR2
      LD R3,SR3
      LD R4,SR4
      LD R5,SR5
      LD R6,SR6
      LD R7,SR7
      BRnzp DELAY
      RET
```

```
;
LINE2 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 PA2_1
      BRz PA2_2
PA2_1 LEA R0,P1
      BRnzp NEXT2
PA2_2 LEA R0,P2
NEXT2 LD R2,SPACE
WRITE LDI R3,DSR
      BRzp WRITE
      STI R2,DDR
      STI R2,DDR
      STI R2,DDR
; write 3 spaces
```

```
        AND R1,R1,#0
        ADD R1,R1,#7
; count down
L2      TRAP x22
        ADD R1,R1,xFFFF
        BRz DONE2
        BRnzp L2
DONE2 LDI R3,DSR
        BRzp DONE
        LD R2,NEWLINE
        STI R2,DDR
        LD R0,SR0
        LD R1,SR1
        LD R2,SR2
        LD R3,SR3
        LD R4,SR4
        LD R5,SR5
        LD R6,SR6
        LD R7,SR7
        BRnzp DELAY
        RET
;
```

```
DELAY ST R1,SR1
        LD R1,COUNT
REP ADD R1,R1,#-1
        BRp REP
        LD R1,SR1
        RET
;
P1 .STRINGZ "**        "
P2 .STRINGZ "##        "
COUNT .FILL x4FFF
NEWLINE .FILL x000A
SPACE .FILL #32
MASK .FILL x4000
DSR .FILL xFE04
DDR .FILL xFE06
KBSR .FILL xFE00
KBDR .FILL xFE02
START .FILL x2000
KB .FILL x0180
SR0 .BLKW 1
SR1 .BLKW 1
SR2 .BLKW 1
```

```
SR3 .BLKW 1
SR4 .BLKW 1
SR5 .BLKW 1
SR6 .BLKW 1
SR7 .BLKW 1
STACK .BLKW 10
; start of actual user program to print the checkerboard
.END
```

interrupt_service_routine:

```
.ORIG x2000
INTERRUPTE      ST R0,SR0
      ST R1,SR1
      ST R2,SR2
      ST R3,SR3
      ST R4,SR3
      ST R5,SR5
      ST R6,SR6
      ST R7,SR7
      LDI R0,KBDR
      LD R1,COUNT
      WRITE LDI R3,DSR
      BRzp WRITE
L     STI R0,DDR
       ADD R1,R1,xFFFF
       BRz DONE
      BRnzp L
DONE LDI R3,DSR
      BRzp DONE
      LD R0,NEWLINE
      STI R0,DDR
      LD R0,SR0
      LD R1,SR1
      LD R2,SR2
      LD R3,SR3
      LD R4,SR4
      LD R5,SR5
      LD R6,SR6
      LD R7,SR7
      ADD R5,R5,#0
      BRz CHANGE1
      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
```