**Tyler Holland**

10/27/09

Lab 4

No Cheating Signature:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TRAP Demo Signature:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Quiz Demo Signature:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 1:**

-The KBSR interrupt is not enabled. At memory location xFE00, the 16 bit number inside is x0000, so the 14th (interrupt) bit is not active because it is a 0.

**Question 2:**

Lines:

x0020 - go to x0021

x0021 - branch to x0052 if last register change was 0

x0052 - go to xFD79

xFD79 - Load xFD7D (Trap x0B) into R0, which loads xFD0B into R0

xFD7A - Load xFD7E (x0000 at this time) to R1

xFD7B - Load xFD7F (Trap x07) into R7, which loads xFD07 into R7

xFD7C - RETurn to R7 which is xFD07

xFD07 - Load xFD3E (x0000) into R0

xFD08 - Load xFD3F (x0053) into R7, which loads x0053 into R7

xFD09 - RETurn to R7 which is x0053

x0053 - go to xFD79 again via Trap x00

It looks like this loop will keep on going until the ready bit in DSR is a 1, and then it give the data to the DDR to be put on the screen. This means it uses polling driven output, because it is looking for the status register to have its bit 15 to change to a 1.

**Question 3:**

.ORIG x3300

ST R7, SaveR7 ;Save R7

ST R2, SaveR2 ;Save R2

ST R3, SaveR3 ;Save R3

AND R3,R3,#0 ;Clear R3

; Start checking for ready

Ready LDI R2, DSR ;Check to see if display is ready

BRzp Ready ;Branch back if display isn't ready (15th bit check)

AND R3,R3,#1 ;Check if R0 has been output

BRp Next ;Print R1

STI R0, DDR ;Put R0 on display

ADD R3,R3,#1 ;Add #1 to R3 so we can print R1

BRnzp Ready ;Loop back to status check

Next STI R1, DDR ;Put R1 on display

; Restore Registers

LD R2, SaveR2 ;Restore R2

LD R3, SaveR3 ;Restore R3

LD R7, SaveR7 ;Restore R7

RET ;End trap

DSR .FILL xFE04

DDR .FILL xFE06

SaveR2 .FILL 0

SaveR3 .FILL 0

SaveR7 .FILL 0

.END

**Question 4:**

;Tyler Holland

;Lab 4 Quiz - What type of cooling you need in your PC

;R0 - Store input/store value

;R1 - Point value of current answer

;R2 - Stores total score

;

.ORIG x3000

AND R1,R1,#0 ;Clear R1

AND R2,R2,#0 ;Clear R2

AND R3,R3,#0 ;Clear R3

; Display Q1

LD R0,Q1 ;Load Q1 address

PUTS ;Display Prompt

JSR GetIn ;Get input

LD R1,Q1A ;Load R1 with Q1 answers

JSR GetVal ;Change to value

ADD R2,R2,R0 ;Add up current total

; Display Q2

LD R0,Q2 ;Load Q2 address

PUTS ;Display Prompt

JSR GetIn ;Get input

LD R1,Q2A ;Load R1 with Q2 answers

JSR GetVal ;Change to value

ADD R2,R2,R0 ;Add up current total

; Display Q3

LD R0,Q3 ;Load Q3 address

PUTS ;Display Prompt

JSR GetIn ;Get input

LD R1,Q3A ;Load R1 with Q3 answers

JSR GetVal ;Change to value

ADD R2,R2,R0 ;Add up current total

; Determine output answer

LD R1,TRY1 ;Put TRY1 in R1

ADD R2,R2,R1 ;First answer determination

BRp SKIP1 ;If it is greater than first answer, continue

LD R0,A1 ;Load Answer 1 in R0

PUTS ;Display answer

BRnzp DONE ;Finish

SKIP1 LD R1,TRY2 ;Put TRY2 in R1

ADD R2,R2,R1 ;Second answer determination

BRp SKIP2 ;If it is greater than second answer, continue

LD R0,A2 ;Load Answer 2 in R0

PUTS ;Display answer

BRnzp DONE ;Finish

SKIP2 LD R1,TRY3 ;Put TRY3 in R1

ADD R2,R2,R1 ;Third answer determination

BRp SKIP3 ;If it is greater than third answer, continue

LD R0,A3 ;Load Answer 3 in R0

PUTS ;Display answer

BRnzp DONE ;Finish

SKIP3 LD R0,A4 ;Load Answer 4 in R0

PUTS ;Display answer

DONE HALT ;End program

;

; Get integer answer function

;

GetIn ST R1, SaveR1 ;Save R1

ST R7, SaveR7 ;Save R7

LEA R0, Prompt ;Load Prompt offset

PUTS ;Print Prompt

;

; User Input stage

;

GETC ;Get a character and put in R0

OUT ;Echo to screen

LD R1,Convert ;Load Convert into R1

ADD R0,R0,R1 ;Convert ASCII to decimal

;

; Reload Registers and return

;

LD R1, SaveR1 ;Reload R1

LD R7, SaveR7 ;Reload R7

RET ;End function

;

; End integer answer function

;

; Start point value function

;

GetVal ADD R0,R0,R1 ;Increment R1 to point to the right answer

LDR R0,R0,#0 ;Load contents of location R0 into R0

RET ;End function

;

; End point value function

;

SaveR1 .FILL 0

SaveR7 .FILL 0

Convert .FILL #-49

Prompt .STRINGZ "Enter your answer: "

;

TRY1 .FILL #-10 ;If 0-10, first answer

TRY2 .FILL #-5 ;If 11-15, second answer

TRY3 .FILL #-6 ;If 16-21, third answer, above that answer 4

Q1 .FILL x3300

Q2 .FILL x336E

Q3 .FILL x33DB

Q1A .FILL x336A

Q2A .FILL x33D7

Q3A .FILL x343D

A1 .FILL x3441

A2 .FILL x3471

A3 .FILL x349F

A4 .FILL x34CE

.END

**DATA:**

.ORIG x3300

Q1 .STRINGZ "\nWhat size is your PC case?\n 1 - Takes up a room\n 2 - Slim/microATX\n 3 - ATX/XL-ATX\n 4 - Mid ATX\n"

Val\_A1\_1 .FILL #7

Val\_A1\_2 .FILL #2

Val\_A1\_3 .FILL #7

Val\_A1\_4 .FILL #4

Q2 .STRINGZ "\nHow many video cards do you have?\n 1 - Integrated\n 2 - One\n 3 - Two or Three\n 4 - Four to Nine\n"

Val\_A2\_1 .FILL #2

Val\_A2\_2 .FILL #4

Val\_A2\_3 .FILL #7

Val\_A2\_4 .FILL #10

Q3 .STRINGZ "\nHow fast is your CPU?\n 1 - Under 2.0ghz\n 2 - 2.0-2.8ghz\n 3 - 2.8ghz-3.6ghz\n 4 - 3.6ghz+\n"

Val\_A3\_1 .FILL #2

Val\_A3\_2 .FILL #4

Val\_A3\_3 .FILL #7

Val\_A3\_4 .FILL #10

Result1 .STRINGZ "\nYou can get by with default cooling, congrats."

Result2 .STRINGZ "\nGet a bunch of fans and water cool your CPU."

Result3 .STRINGZ "\nYou should have multiple water cooling loops."

Result4 .STRINGZ "\nConsider using Liquid Nitrogen before your CPU explodes!"

.END