.global \_start

\_start:

ldr r0, =LED

ldr r1, =switch

ldr r9, =SSD

ldr r10, =table

main:

// Clean R4 and R5 registers

mov r4, #0

mov r5, #0

// Load data from switches, store to LEDs

ldr r2, [r1]

str r2, [r0]

// Conversion of binary to BCD done using double dabble algorithm

// Move R2 to R3, do initial 2 shifts, set counter in R8 to 2

mov r3, r2

mov r3, r3, lsl #2

mov r8, #2

loop:

// Do a shift, increment counter

mov r3, r3, lsl #1

add r8, #1

// Check whether 8 shifts have been done, if so, jump to display

cmp r8, #0x8

beq display

// Colums are saved in R4 register and compared to 0x500, 0x5000 and 0x50000

// Check if ones column is larger than 5, add 3 if it is

and r4, r3, #0xF00

cmp r4, #0x500

addge r3, r3, #0x300

// Check if tens column is larger than 5, add 3 if it is

and r4, r3, #0xF000

cmp r4, #0x5000

addge r3, r3, #0x3000

// Check if hundreds column is larger than 5, add 3 if it is

and r4, r3, #0xF0000

cmp r4, #0x50000

addge r3, r3, #0x30000

// Loop to beginning

b loop

// Digits are converted to SSD hex code and saved in R5 register one by one

display:

// Load first digit, convert it to SSD hex code, save to R5

// Digit is loaded from R3 into R4, right shifted 16 bytes so we get just the first digit

mov r4, r3, lsr #16

ldrb r4, [r10, r4]

// Save first digit in R5, left shifted 16 bytes

add r5, r4, lsl #16

// Load 2nd digit, convert it to SSD hex code, save to R5

// Digit is loaded from R4 into R4, right shifted 12 bytes and AND'd with 0xF to get just the 2nd digit

mov r4, r3, lsr #12

and r4, #0x0F

ldrb r4, [r10, r4]

// Save second digit in R5, left shifted 8 bytes

add r5, r4, lsl #8

// Load 3rd digit, convert it to SSD hex code, save to R5

// Digit is loaded from R4 into R4, right shifted 8 bytes and AND'd with 0xF to get just the 3rd digit

mov r4, r3, lsr #8

and r4, #0x0F

ldrb r4, [r10, r4]

// Save second digit in R5

add r5, r4

// Store final result to SSD address

str r5, [r9]

// Jump to program start

b main

.data

.equ LED, 0xFF200000

.equ switch, 0xFF200040

.equ SSD, 0xFF200020

// Table of SSD values, 0-9

table:

.byte 0x3F

.byte 0x06

.byte 0x5B

.byte 0x4F

.byte 0x66

.byte 0x6D

.byte 0x7D

.byte 0x07

.byte 0x7F

.byte 0x6F

.byte 0x77