아래는 csd_asm.S 코드입니다.

```
#define csd_SWITCH 0x41210000

#include "csd_zynq_peripherals.h"

.section .csd_boot,"ax"

.extern csd_main

.align 8
// Our interrupt vector table
csd_entry:
        b csd_reset
        b .
        b .
        b .
        b .
        b .
        b csd_irq
        b .

.global main
csd_reset:
main:

        // Set VBAR (Vector Base Address Register) to my own interrupt vectors
    ldr     r0, =csd_entry
    mcr      p15, 0, r0, c12, c0, 0
    dsb
    isb

        // Read Cache Type Register (CTR)
        mrc p15, 0, r1, c0, c0, 1

        // Read Cache Level ID Register (CLIDR)
        mrc p15, 1, r2, c0, c0, 1

        ldr r3, =csd_SWITCH // save the address of switch in r3
        ldr r4, [r3] // load the status of switch in r4

        and r4, r4, #1 // check if sw0 is on or off
        cmp r4, #1
        beq cache_on // if sw0 is on, caches are enabled
        bne cache_off // if sw0 is off, caches are disabled

cache_on:

        @-----------------------
        @ Enable Caches (L2)
        @-----------------------
        ldr r0, =L2_reg1_ctrl
    mov r1, #0x1
    str r1, [r0]

        @-----------------------
        @ Enable Caches (IL1, DL1)
        @-----------------------
        mrc             p15, 0, r0, c1, c0, 0    @ read control register (CP15
register1)
        orr             r0, r0, #(1<<12)     @ Enable I bit (Instruction Cache)
        orr             r0, r0, #(1<<2)          @ Enable C bit (Data and Unified
Caches)
        mcr             p15, 0, r0, c1, c0, 0    @ write control register (CP15
```

```
register2)

        b forever

cache_off:

        @-----------------------
        @ Disable Caches (L2)
        @-----------------------
        ldr r0, =L2_reg1_ctrl
    mov r1, #0x0
    str r1, [r0]
        @-----------------------
        @ Disable Caches (IL1, DL1)
        @-----------------------
        mrc             p15, 0, r0, c1, c0, 0    @ read control register (CP15
register1)
        bic             r0, r0, #4096                @ disable I bit (Instruction
Cache)
        bic             r0, r0, #4                       @ disable C bit (Data and
Unified Caches)
        mcr             p15, 0, r0, c1, c0, 0    @ write control register (CP15
register2)

        // read SCTLR (System Control Register) to r0
        mrc     p15, 0, r0, c1, c0, 0

        b csd_main

forever:
        nop
        b forever

// Normal Interrupt Service Routine
csd_irq:
        b .

.end
```

아래는 csd_main.c 코드입니다.

```
/*
 * csd_main.c
 *
 *  Created on: 2018. 4. 30.
 *      Author: Taeweon Suh
 */

unsigned volatile char * gpio_led = (unsigned char *) 0x41200000;

int csd_main()
{

 int count;

 while (1) {

        for (count=0; count < 0x400000; count++) ;

        *gpio_led = 0xFF;

        for (count=0; count < 0x400000; count++) ;

        *gpio_led = 0x0;

 }
        return 0;
}
```