Hardware/Software Codesign Lab 5

Student Name: Jose Sotelo

Student ID: 013969681

1. Follow the Lab 5 manual to finish Lab 5 and perform the following two demonstrations to your instructor:
   1) Program FPGA and download software application to the board to verify operations on hardware.
   2) Demonstrate step 4: Launch Debugger and debug

2. Copy and paste the following information to the end of this document:
   1) Lab5.c

```
#include "xparameters.h"
#include "xgpio.h"
#include "led_ip.h"
// Include xscutimer header file
#include "xscutimer.h"

//====================================================
XScuTimer Timer;              /* Cortex A9 SCU Private Timer Instance */

#define ONE_TENTH 32500000 // half of the CPU clock speed/10

int main (void)
{
   XGpio dip, push;
   int psb_check, dip_check, dip_check_prev, count, Status;

   // PS Timer related definitions
   XScuTimer_Config *ConfigPtr;
   XScuTimer *TimerInstancePtr = &Timer;

   xil_printf("-- Start of the Program --\r\n");

   XGpio_Initialize(&dip, XPAR_SWITCHES_DEVICE_ID);
   XGpio_SetDataDirection(&dip, 1, 0xffffffff);

   XGpio_Initialize(&push, XPAR_BUTTONS_DEVICE_ID);
   XGpio_SetDataDirection(&push, 1, 0xffffffff);

   count = 0;

   // Initialize the timer
   ConfigPtr = XScuTimer_LookupConfig(XPAR_PS7_SCUTIMER_0_DEVICE_ID);
   Status = XScuTimer_CfgInitialize(TimerInstancePtr, ConfigPtr, ConfigPtr-
>BaseAddr);

   if(Status != XST_SUCCESS)
   {
           xil_printf("Timer init() failed\r\n");
           return XST_FAILURE;
   }

   // Read dip switch values
   dip_check_prev = XGpio_DiscreteRead(&dip, 1);

   // Load timer with delay in multiple of ONE_TENTH
   XScuTimer_LoadTimer(TimerInstancePtr, ONE_TENTH*dip_check_prev);

   // Set AutoLoad mode
   XScuTimer_EnableAutoReload(TimerInstancePtr);
```

```
50
51    // Start the timer
52    XScuTimer_Start(TimerInstancePtr);
53
54    while (1)
55    {
56           // Read push buttons and break the loop if Center button pressed
57           psb_check = XGpio_DiscreteRead(&push, 1);
58
59           if(psb_check > 0)
60           {
61                  xil_printf("Push button pressed: Exiting\r\n");
62                  XScuTimer_Stop(TimerInstancePtr);
63                  break;
64           }
65
66           dip_check = XGpio_DiscreteRead(&dip, 1);
67
68           if (dip_check != dip_check_prev)
69           {
70                  xil_printf("DIP Switch Status %x, %x\r\n", dip_check_prev,
71 dip_check);
72                  dip_check_prev = dip_check;
73
74                  // load timer with the new switch settings
75                  XScuTimer_LoadTimer(TimerInstancePtr,
76 ONE_TENTH*dip_check);
77                  count = 0;
78           }
79
80           if(XScuTimer_IsExpired(TimerInstancePtr))
81           {
82                        // clear status bit
83                        XScuTimer_ClearInterruptStatus(TimerInstancePtr);
84
85                        // output the count to LED and increment the count
86                        LED_IP_mWriteReg(XPAR_LED_IP_S_AXI_BASEADDR, 0,
87 count);
88                        count++;
           }
     }
     return 0;
  }
```

3. Answer the following questions:
   1) What is the prescale value of the private timer used in this lab?

```
#define XSCUTIMER_CONTROL_PRESCALER_MASK      0x0000FF00U
```

   2) What is the minimum time interval and maximum time interval controlled by the dip switch in this lab? Please show your calculation.

```
#define XPAR_PS7_CORTEXA9_0_CPU_CLK_FREQ_HZ 666666687
```

      Minimum = 1*((0.5*666666687)/10)

      Maximum = 15*((0.5*666666687)/10)

   3) List timer driver calling sequence.

      1. Add the include file “xscutimer.h”

      2. Add PS timer related definitions.

      3. Initialize the timer using the XScuTimer_LookUpConfig and XScuTimer_CfgInitialize function

      4. Load timer with delay using the XScuTimer_LoadTimer function

      5. Set AutoLoad mode using the XScuTime_EnableAutoReload function

      6. Start the timer using the XScuTimer_Start function

      7. Load timer with a new setting depending on switch or dip switch using the XScuTimer_LoadTimer function

      8. Check for when the timer has expired using the XScuTimer_IsExpired function

      9. Clear the status bit using the XScuTimer_ClearInterruptStatus function