#define SWITCH1 0x50

#define SWITCH2 0x40

#define SWITCH3 0x30

#define SWITCH4 0x20

#include <xparameters.h>

#include <xgpio.h>

#define WAIT\_VAL 0x1000000 // iterations to wait in delay

#define SWITCH1 0x50

#define SWITCH2 0x40

#define SWITCH3 0x30

#define SWITCH4 0x20

#define NORTH 0x10

#define SOUTH 0x8

#define EAST 0x4

#define WEST 0x2

#define CENTER 0x1

#define true 1

#define false 0

#define bool \_Bool // bool not defined in old c std, \_Bool is

int delay(void);

int main(){

int count=0; //init to 0

int count\_masked=0;

//true for right/east button/ dip status mode

bool dip\_mode=false;//false for left/west button/counter mode

int prev=0;//determine when to show update of switches

//Set up Hardware

XGpio led; //Hardware io

XGpio inputs;//switches 0:3 and buttons 4:8

XGpio\_Initialize(&led, XPAR\_LEDS\_DEVICE\_ID);

XGpio\_SetDataDirection(&led,1,0); //setting led direction to output

XGpio\_Initialize(&inputs, XPAR\_INPUTS\_DEVICE\_ID);

XGpio\_SetDataDirection(&inputs,1,1); //setting direction ro input

while(1) //infinite loop to keep circuit always executing

{

//int debug\_inf=XGpio\_DiscreteRead(&inputs,1);

// xil\_printf("Looping 0x%x\n\r",debug\_inf); //Print count to console

if ((CENTER & XGpio\_DiscreteRead(&inputs,1))==CENTER){

count=0;//RESET count

xil\_printf("RESET: 0x%x\n\r", count); //Print count to console

}

if ((WEST & XGpio\_DiscreteRead(&inputs,1))==WEST){

dip\_mode=false;//set mode to count

xil\_printf("mode switch: Counter 0x%x\n\r", count\_masked); //Print count to console

}

else if ((EAST & XGpio\_DiscreteRead(&inputs,1))==EAST){

dip\_mode=true;//set mode to dip status

xil\_printf("mode switch: DIP 0x%x\n\r", count\_masked); //Print count to console

}

if (!dip\_mode){

if ((NORTH & XGpio\_DiscreteRead(&inputs,1))==NORTH){ //& operation masks non-NORTH bits

count++;

count\_masked = count & 0xf;//get lower 4 bits of count, so that count masked rolls over every 16 counts

XGpio\_DiscreteWrite(&led,1,count\_masked);//write count masked to leds

xil\_printf("incremented 0x%x\n\r",count\_masked); //Print count to console

}

else if((SOUTH & XGpio\_DiscreteRead(&inputs,1))==SOUTH){

count--;

count\_masked = count & 0xf;//get lower 4 bits of count, so that count masked rolls over every 16 counts

XGpio\_DiscreteWrite(&led,1,count\_masked);//write count masked to leds

xil\_printf("decremented 0x%x\n\r",count\_masked); //Print count to console

}

}

else{

count\_masked = 0x1f0 & XGpio\_DiscreteRead(&inputs,1); //takes upper bits of switches

count\_masked = count\_masked>>5;

if (prev!=count\_masked){

XGpio\_DiscreteWrite(&led,1,count\_masked);//writes value of switches

xil\_printf("Switches changed 0x%x \n\r",count\_masked); //Print count to console

}

prev=count\_masked;

}

delay();

}

return (0);

}

int delay (void)

{

volatile int delay\_count=0; // volatile prevents compiler optimization

while(delay\_count<WAIT\_VAL)

delay\_count++;// each iteration of while loop is 2 clock cycles

return(0);

}