/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* Copyright (C) 2009 - 2014 Xilinx, Inc. All rights reserved.

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy

\* of this software and associated documentation files (the "Software"), to deal

\* in the Software without restriction, including without limitation the rights

\* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

\* copies of the Software, and to permit persons to whom the Software is

\* furnished to do so, subject to the following conditions:

\*

\* The above copyright notice and this permission notice shall be included in

\* all copies or substantial portions of the Software.

\*

\* Use of the Software is limited solely to applications:

\* (a) running on a Xilinx device, or

\* (b) that interact with a Xilinx device through a bus or interconnect.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

\* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

\* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL

\* XILINX BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,

\* WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF

\* OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

\* SOFTWARE.

\*

\* Except as contained in this notice, the name of the Xilinx shall not be used

\* in advertising or otherwise to promote the sale, use or other dealings in

\* this Software without prior written authorization from Xilinx.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\* helloworld.c: simple test application

\*

\* This application configures UART 16550 to baud rate 9600.

\* PS7 UART (Zynq) is not initialized by this application, since

\* bootrom/bsp configures it to baud rate 115200

\*

\* ------------------------------------------------

\* | UART TYPE BAUD RATE |

\* ------------------------------------------------

\* uartns550 9600

\* uartlite Configurable only in HW design

\* ps7\_uart 115200 (configured by bootrom/bsp)

\*/

#include <stdio.h>

#include "platform.h"

#include "xil\_printf.h"

#include <unistd.h>

#define GPIO\_BTN\_ADDR 0x40000000

#define GPIO\_PADDLE\_L\_ADDR 0x40000004

#define GPIO\_PADDLE\_R\_ADDR 0x40000008

#define GPIO\_BALL\_X\_ADDR 0x4000000C

#define GPIO\_BALL\_Y\_ADDR 0x40000010

#define GPIO\_SCORE\_LEFT\_ADDR 0x40000014

#define GPIO\_SCORE\_RIGHT\_ADDR 0x40000018

#define GPIO\_RESET\_ADDR 0x4000001C

#define GPIO\_LED\_ADDR 0x40000020

#define GPIO\_DEVIL\_ADDR 0x40000024

#define GPIO\_START\_ADDR 0x40000028

int main()

{

uint32\_t \* btn = (uint32\_t \*) GPIO\_BTN\_ADDR;

uint32\_t \* pad\_l = (uint32\_t \*) GPIO\_PADDLE\_L\_ADDR;

uint32\_t \* pad\_r = (uint32\_t \*) GPIO\_PADDLE\_R\_ADDR;

uint32\_t \* ball\_x = (uint32\_t \*) GPIO\_BALL\_X\_ADDR;

uint32\_t \* ball\_y = (uint32\_t \*) GPIO\_BALL\_Y\_ADDR;

uint32\_t \* score\_l = (uint32\_t \*) GPIO\_SCORE\_LEFT\_ADDR;

uint32\_t \* score\_r = (uint32\_t \*) GPIO\_SCORE\_RIGHT\_ADDR;

uint32\_t \* reset = (uint32\_t \*) GPIO\_RESET\_ADDR;

uint32\_t \* led = (uint32\_t \*) GPIO\_LED\_ADDR;

uint32\_t \* devil = (uint32\_t \*) GPIO\_DEVIL\_ADDR;

uint32\_t \* start = (uint32\_t \*) GPIO\_START\_ADDR;

//int paddle\_speed;

//int ball\_speed;

int paddle\_length = 80;

int paddle\_width = 20;

long prescaler\_ball;

long prescaler\_paddle;

long counter = 0;

int ball\_r = 20;

int updown = 0; //0: up, 1: down

int leftright = 1;//0: left, 1: down

int paddle\_l\_up = 0; //0: up, 1: down

int paddle\_l\_down = 0;

int paddle\_r\_up = 0; // 0: up, 1: down

int paddle\_r\_down = 0;

init\_platform();

pad\_l[0] = 260;

pad\_r[0] = 260;

ball\_x[0] = 390;

ball\_y[0] = 290;

while(1){

if(devil[0] == 1){ //normal mode switch3 T16

prescaler\_ball = 2000;

prescaler\_paddle = 500;

counter = 0;

}

if(devil[0] == 0){ //expert mode

prescaler\_ball = 800;

prescaler\_paddle = 500;

counter = 0;

}

while(start[0] == 1){ // switch 2 W13

if((ball\_x[0] + ball\_r) > 790){

score\_l[0] = score\_l[0] + 1;

pad\_l[0] = 260;

pad\_r[0] = 260;

ball\_x[0] = 390;

ball\_y[0] = 290;

leftright = 1;

}

if((ball\_x[0]) < 10){

score\_r[0] = score\_r[0] + 1;

pad\_l[0] = 260;

pad\_r[0] = 260;

ball\_x[0] = 390;

ball\_y[0] = 290;

leftright = 0;

}

led[0] = btn[0];

if(reset[0] == 1){

pad\_l[0] = 260;

pad\_r[0] = 260;

ball\_x[0] = 390;

ball\_y[0] = 290;

score\_l[0] = 0;

score\_r[0] = 0;

}

if(score\_l[0] == 10 || score\_r[0] == 10){

pad\_l[0] = 260;

pad\_r[0] = 260;

ball\_x[0] = 390;

ball\_y[0] = 290;

score\_l[0] = 0;

score\_r[0] = 0;

sleep(10);

}

if(btn[0] & 0b0001){

paddle\_r\_down = 1;

}

if(btn[0] & 0b0010){

paddle\_r\_up = 1;

}

if(btn[0] & 0b0100){

paddle\_l\_down = 1;

}

if(btn[0] & 0b1000){

paddle\_l\_up = 1;

}

if(btn[0] == 0b0000){

paddle\_l\_up = 0; //0: up, 1: down

paddle\_l\_down = 0;

paddle\_r\_up = 0; // 0: up, 1: down

paddle\_r\_down = 0;

}

if(counter == prescaler\_paddle){

//make the paddles moving

if(paddle\_r\_down == 1){

pad\_r[0] = pad\_r[0] + 1;

if(pad\_r[0] >= 590 - paddle\_length){

pad\_r[0] = 590 - paddle\_length;

}

}

if(paddle\_r\_up == 1){

pad\_r[0] = pad\_r[0] - 1;

if(pad\_r[0] <= 10){

pad\_r[0] = 10;

}

}

if(paddle\_l\_down == 1){

pad\_l[0] = pad\_l[0] + 1;

if(pad\_l[0] >= 590 - paddle\_length){

pad\_l[0] = 590 - paddle\_length;

}

}

if(paddle\_l\_up == 1){

pad\_l[0] = pad\_l[0] - 1;

if(pad\_l[0] <= 10){

pad\_l[0] = 10;

}

}

}

else{

counter++;

}

if(counter == prescaler\_ball){

//move the ball

if(ball\_y[0] > 10 && updown == 0){ //up

ball\_y[0] = ball\_y[0] - 1;

}

if(ball\_y[0] < 570 && updown == 1){ //down

ball\_y[0] = ball\_y[0] + 1;

}

if(leftright == 0){ //left

ball\_x[0] = ball\_x[0] - 1;

}

if(leftright == 1){ //right

ball\_x[0] = ball\_x[0] + 1;

}

if(ball\_y[0] == 10 && updown == 0){

updown = 1;

}

if(ball\_y[0] == 570 && updown == 1){ //down

updown = 0;

}

if(((ball\_x[0] + ball\_r) >= (790 - paddle\_width)) && (ball\_y[0] + ball\_r > pad\_r[0]) && (ball\_y[0] < (pad\_r[0] + paddle\_length))){

leftright = 0;

}

if((ball\_x[0] <= (10 + paddle\_width)) && ((ball\_y[0] + ball\_r) > pad\_l[0]) && (ball\_y[0] < (pad\_l[0] + paddle\_length))){

leftright = 1;

}

counter = 0;

}

else{

counter ++;

}

}

pad\_l[0] = 260;

pad\_r[0] = 260;

ball\_x[0] = 390;

ball\_y[0] = 290;

score\_l[0] = 0;

score\_r[0] = 0;

}

cleanup\_platform();

return 0;

}