```c
/*
 * sound.c
 * Taylor Cowley and Andrew Okazaki
 */

#include <stdint.h>
#include <stdbool.h>
#include "xac97_l.h"
#include "xparameters.h"
#define ZERO 0
#define INCREMENTAL_STEP 100
#define SHIFT 16
#define VOLUME_STEP 30          // Volume up or down by 30 each time
//tank-------------------------------------------------------------------------
extern int32_t tankFireSoundRate;
extern int32_t tankFireSoundFrames;
extern int32_t tankFireSound[];
struct sound_tank{
    int32_t count;
}sound_tank;

//Explode----------------------------------------------------------------------
extern int32_t tankExplodeSoundRate;
extern int32_t tankExplosionSoundFrames;
extern int32_t tankExplosionSound[];
struct sound_explosion{
    int32_t count;
}sound_explosion;

//Alien sound 1----------------------------------------------------------------
extern int32_t alien1SoundRate;
extern int32_t alien1SoundFrames;
extern int32_t alien1Sound[];
struct sound_alien1{
    int32_t count;
}sound_alien1;

//Alien sound 2----------------------------------------------------------------
extern int32_t alien2SoundRate;
extern int32_t alien2SoundFrames;
extern int32_t alien2Sound[];
struct sound_alien2{
    int32_t count;
}sound_alien2;

//Alien sound 3----------------------------------------------------------------
extern int32_t alien3SoundRate;
extern int32_t alien3SoundFrames;
extern int32_t alien3Sound[];
struct sound_alien3{
    int32_t count;
}sound_alien3;

//Alien sound 4----------------------------------------------------------------
extern int32_t alien4SoundRate;
extern int32_t alien4SoundFrames;
extern int32_t alien4Sound[];
struct sound_alien4{
```

```
    int32_t count;
}sound_alien4;

//Alien Kill-------------------------------------------------------------------
extern int32_t alienKillSoundRate;
extern int32_t alienKillSoundFrames;
extern int32_t alienKillSound[];
struct sound_alienKill{
    int32_t count;
}sound_alienKill;

//Mother Ship High-------------------------------------------------------------
extern int32_t motherShipHighSoundRate;
extern int32_t motherShipHighSoundFrames;
extern int32_t motherShipHighSound[];
struct sound_motherShipHigh{
    int32_t count;
}sound_motherShipHigh;

//Mother ship Low--------------------------------------------------------------
extern int32_t motherShipLowSoundRate;
extern int32_t motherShipLowSoundFrames;
extern int32_t motherShipLowSound[];
struct sound_motherShipLow{
    int32_t count;
}sound_motherShipLow;

//Global volume control
uint32_t sound_vol;

void sound_init_AC_97(){
    //Reset the ac97 ---------------------------------------------------------
    XAC97_HardReset(XPAR_AXI_AC97_0_BASEADDR);

    //Wait-------------------------------------------------------------------
    XAC97_AwaitCodecReady(XPAR_AXI_AC97_0_BASEADDR);

    //enable VRA-------------------------------------------------------------
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_ExtendedAudioStat\
            ,AC97_EXTENDED_AUDIO_CONTROL_VRA);

    //set frequency----------------------------------------------------------
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_PCM_DAC_Rate\
            , AC97_PCM_RATE_8000_HZ);

    //set volumes------------------------------------------------------------
    sound_vol = AC97_VOL_MID;
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_MasterVol, sound_vol);
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_AuxOutVol, sound_vol);
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_MasterVolMono, sound_vol);
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_PCBeepVol, sound_vol);
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_PCMOutVol, sound_vol);
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_LineInVol, sound_vol);
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_MicVol, sound_vol);

    //clear fifos------------------------------------------------------------
    XAC97_ClearFifos(XPAR_AXI_AC97_0_BASEADDR);
```

```
    //set up interrupts--------------------------------------------------------
    XAC97_mSetControl(XPAR_AXI_AC97_0_BASEADDR,AC97_ENABLE_IN_FIFO_INTERRUPT);
}

// initialize the tank struct
void sound_init_tank(){
    sound_tank.count = ZERO; // initialize to 0
}
// initialize the explosion struct
void sound_init_explosion(){
    sound_explosion.count = ZERO;
}
// initialize alien1 struct
void sound_init_alien1(){
    sound_alien1.count = ZERO;// initialize to 0
}
// initialize alien2 struct
void sound_init_alien2(){
    sound_alien2.count = ZERO;// initialize to 0
}
// initialize alien3 struct
void sound_init_alien3(){
    sound_alien3.count = ZERO;// initialize to 0
}
// initialize alien4 struct
void sound_init_alien4(){
    sound_alien4.count = ZERO;// initialize to 0
}
// initialize the alien Kill struct
void sound_init_alienKill(){
    sound_alienKill.count = ZERO;// initialize to 0
}
// initialize the Mother Ship High freq struct
void sound_init_motherShipHigh(){
    sound_motherShipHigh.count = ZERO;// initialize to 0
}
// initialize the Mother Ship Low freq struct
void sound_init_motherShipLow(){
    // If the count is going and has not ended do not re initialize
    if(sound_motherShipLow.count >= motherShipLowSoundFrames){
        sound_motherShipLow.count = ZERO;// initialize to 0
    }
}


//Play sound of tank shooting
void sound_func(int32_t SoundRate, int32_t SoundFrames,int32_t Sound[], int32_t count){
    uint32_t i; // initiate variable
    uint32_t sample = 0; // initiate variable to 0
    // Get the sound card all set up with the sound rate
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR,AC97_PCM_DAC_Rate,SoundRate);
    for(i = 0; i < INCREMENTAL_STEP; i++){  // Gives the sound card
        count ++;
        sample = (Sound[count] << SHIFT) | tankFireSound[count];
        XAC97_mSetInFifoData(XPAR_AXI_AC97_0_BASEADDR, sample); // give sample to sound
card
        if(count >= tankFireSoundFrames){   // resets the fifo
            XAC97_ClearFifos(XPAR_AXI_AC97_0_BASEADDR);
```

```
            XAC97_mSetControl(XPAR_AXI_AC97_0_BASEADDR,AC97_ENABLE_IN_FIFO_INTERRUPT);
        }
    }
}

//Play sound of tank shooting
void tank_sound_func(){
    uint32_t i; // initiate variable
    uint32_t sample = 0; // initiate variable to 0
    // set up sound card with sound rate
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR,AC97_PCM_DAC_Rate,tankFireSoundRate);
    for(i = 0; i < INCREMENTAL_STEP; i++){  // give it 100 samples
        sound_tank.count ++;
        // create a sample; give it to sound card
        sample = (tankFireSound[sound_tank.count] << SHIFT) |
tankFireSound[sound_tank.count] ;
        XAC97_mSetInFifoData(XPAR_AXI_AC97_0_BASEADDR, sample);
        if(sound_tank.count >= tankFireSoundFrames){// reset it at end of sound
            XAC97_ClearFifos(XPAR_AXI_AC97_0_BASEADDR);
            XAC97_mSetControl(XPAR_AXI_AC97_0_BASEADDR,AC97_ENABLE_IN_FIFO_INTERRUPT);
        }
    }
}
//Play sound of Mother Ship moving across the screen
void ufo_sound_func(){
    uint32_t i; // initiate variable
    uint32_t sample = 0; // initiate variable to 0
    // Set up sound card with proper sound rate
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_PCM_DAC_Rate, motherShipLowSoundRate);
    for(i = 0; i < INCREMENTAL_STEP; i++){
        sound_motherShipLow.count ++;   // increment
        // Create sample and give it to FIFO
        sample = (motherShipLowSound[sound_motherShipLow.count] << SHIFT) |
motherShipLowSound[sound_motherShipLow.count] ;
        XAC97_mSetInFifoData(XPAR_AXI_AC97_0_BASEADDR, sample);
        if(sound_motherShipLow.count >= motherShipLowSoundFrames){//    At end of sound,
reset.
            XAC97_ClearFifos(XPAR_AXI_AC97_0_BASEADDR);
            XAC97_mSetControl(XPAR_AXI_AC97_0_BASEADDR,AC97_ENABLE_IN_FIFO_INTERRUPT);
        }
    }
}

// Play sound for the kill Mother Ship
void ufo_kill_sound_func(){
    uint32_t i; // initiate variable
    uint32_t sample = 0; // initiate variable to 0
    // Give sound card proper sound rate
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_PCM_DAC_Rate, motherShipHighSoundRate);
    for(i = 0; i < INCREMENTAL_STEP; i++){
        sound_motherShipHigh.count ++; // increment
        // Make sample and give it to FIFO
        sample = (motherShipHighSound[sound_motherShipHigh.count] << SHIFT) |
motherShipHighSound[sound_motherShipHigh.count] ;
        XAC97_mSetInFifoData(XPAR_AXI_AC97_0_BASEADDR, sample);
        if(sound_motherShipHigh.count >= motherShipHighSoundFrames){// At end of sound,
reset
            XAC97_ClearFifos(XPAR_AXI_AC97_0_BASEADDR);
```

```
            XAC97_mSetControl(XPAR_AXI_AC97_0_BASEADDR,AC97_ENABLE_IN_FIFO_INTERRUPT);
        }
    }
}

//Play sound for when the tank explodes
void tank_explosion_sound_func(){
    uint32_t i; // initiate variable
    uint32_t sample = 0; // initiate variable to 0
    // Give sound card proper sound rate
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_PCM_DAC_Rate, tankExplodeSoundRate);
    for(i = 0; i < INCREMENTAL_STEP; i++){
        sound_explosion.count ++;   // increment
        sample = (tankExplosionSound[sound_explosion.count] << SHIFT) |
tankExplosionSound[sound_explosion.count] ;
        XAC97_mSetInFifoData(XPAR_AXI_AC97_0_BASEADDR, sample);
        if(sound_explosion.count >= tankExplosionSoundFrames){// at end of sound, reset
            XAC97_ClearFifos(XPAR_AXI_AC97_0_BASEADDR);
            XAC97_mSetControl(XPAR_AXI_AC97_0_BASEADDR,AC97_ENABLE_IN_FIFO_INTERRUPT);
        }
    }
}


//Run function checking all sounds and then play sounds
void sound_run(){
    uint32_t i; // initiate variable
    uint32_t sample = 0; // initiate variable to 0

    // This block of code sends sound data of only one sound at a time
    if(sound_tank.count <= tankFireSoundFrames){                    // Tank fire
        tank_sound_func();
    }else if(sound_motherShipLow.count <= motherShipLowSoundFrames){// Mother ship 1
        ufo_sound_func();
    }else if(sound_motherShipHigh.count <= motherShipHighSoundFrames){// Mother ship 2
        ufo_kill_sound_func();
    }else if(sound_explosion.count <= tankExplosionSoundFrames){// tank death
        tank_explosion_sound_func();
    }else if(sound_alien1.count <= alien1SoundFrames){          // Alien 1
        // Give sound card good sound rate
        XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_PCM_DAC_Rate, alien1SoundRate);
        for(i = 0; i < INCREMENTAL_STEP; i++){  // 100 samples
            sound_alien1.count ++;
            // Create sample and load it into FIFO
            sample = (alien1Sound[sound_alien1.count] << SHIFT) |
alien1Sound[sound_alien1.count] ;
            XAC97_mSetInFifoData(XPAR_AXI_AC97_0_BASEADDR, sample);
            if(sound_alien1.count >= alien1SoundFrames){    // end of sound; reset
                XAC97_ClearFifos(XPAR_AXI_AC97_0_BASEADDR);
                XAC97_mSetControl(XPAR_AXI_AC97_0_BASEADDR,AC97_ENABLE_IN_FIFO_INTERRUPT);
            }
        }
    }else if(sound_alien2.count <= alien2SoundFrames){          // Alien 2
        // Give sound card good sample rate
        XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_PCM_DAC_Rate, alien2SoundRate);
        for(i = 0; i < INCREMENTAL_STEP; i++){
            sound_alien2.count ++;
            // Create sample and give it to FIFO
```

```
            sample = (alien2Sound[sound_alien2.count] << SHIFT) |
alien2Sound[sound_alien2.count] ;
            XAC97_mSetInFifoData(XPAR_AXI_AC97_0_BASEADDR, sample);
            if(sound_alien2.count >= alien2SoundFrames){    // end of sound, reset
                XAC97_ClearFifos(XPAR_AXI_AC97_0_BASEADDR);
                XAC97_mSetControl(XPAR_AXI_AC97_0_BASEADDR,AC97_ENABLE_IN_FIFO_INTERRUPT);
            }
        }
    }else if(sound_alien3.count <= alien3SoundFrames){          // Alien 3
        // Give sound card good sample rate
        XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_PCM_DAC_Rate, alien3SoundRate);
        for(i = 0; i < INCREMENTAL_STEP; i++){
            sound_alien3.count ++;
            // Create sample and give it to sound card
            sample = (alien2Sound[sound_alien3.count] << SHIFT) |
alien2Sound[sound_alien3.count] ;
            XAC97_mSetInFifoData(XPAR_AXI_AC97_0_BASEADDR, sample);
            if(sound_alien3.count >= alien3SoundFrames){    // end of sound, reset
                XAC97_ClearFifos(XPAR_AXI_AC97_0_BASEADDR);
                XAC97_mSetControl(XPAR_AXI_AC97_0_BASEADDR,AC97_ENABLE_IN_FIFO_INTERRUPT);
            }
        }
    }else if(sound_alien4.count <= alien4SoundFrames){          // Alien 4
        // Give sound card good sample rate
        XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_PCM_DAC_Rate, alien4SoundRate);
        for(i = 0; i < INCREMENTAL_STEP; i++){
            sound_alien4.count ++;
            // Create sample and load it into FIFO
            sample = (alien4Sound[sound_alien4.count] << SHIFT) |
alien4Sound[sound_alien4.count] ;
            XAC97_mSetInFifoData(XPAR_AXI_AC97_0_BASEADDR, sample);
            if(sound_alien4.count >= alien4SoundFrames){    // end of sound, reset
                XAC97_ClearFifos(XPAR_AXI_AC97_0_BASEADDR);
                XAC97_mSetControl(XPAR_AXI_AC97_0_BASEADDR,AC97_ENABLE_IN_FIFO_INTERRUPT);
            }
        }
    }else if(sound_alienKill.count <= alienKillSoundFrames){    // ALien death
        // Give sound card good sound rate
        XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_PCM_DAC_Rate, alienKillSoundRate);
        for(i = 0; i < INCREMENTAL_STEP; i++){
            sound_alienKill.count ++;
            // Create sample, give it to sound card
            sample = (alienKillSound[sound_alienKill.count] << SHIFT) |
alienKillSound[sound_alienKill.count] ;
            XAC97_mSetInFifoData(XPAR_AXI_AC97_0_BASEADDR, sample);
            if(sound_alienKill.count >= alienKillSoundFrames){  // end of sample, reset
                XAC97_ClearFifos(XPAR_AXI_AC97_0_BASEADDR);
                XAC97_mSetControl(XPAR_AXI_AC97_0_BASEADDR,AC97_ENABLE_IN_FIFO_INTERRUPT);
            }
        }
    }else{                                              // No sound playing; clear.
        XAC97_ClearFifos(XPAR_AXI_AC97_0_BASEADDR);
        XAC97_mSetControl(XPAR_AXI_AC97_0_BASEADDR,AC97_ENABLE_IN_FIFO_INTERRUPT);
    }
}

// turn the sound up
void sound_vol_up(){
```

```
        sound_vol += VOLUME_STEP;
        XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_AuxOutVol, sound_vol);
}

// turn the sound down
void sound_vol_down(){
    sound_vol -= VOLUME_STEP;
    XAC97_WriteReg(XPAR_AXI_AC97_0_BASEADDR, AC97_AuxOutVol, sound_vol);
}

// Stop playing the mother ship sound
void sound_motherShipStop(){
    // Set it to the end
    sound_motherShipLow.count = motherShipLowSoundFrames+ VOLUME_STEP;
}
```