#include <stdio.h>

#include "reg\_setup.h"

#define RXSR\_OFFSET 0

typedef struct {

char \*name;

int start\_bit;

int size\_bits;

unsigned int stored\_value;

} rxsr\_field\_t;

rxsr\_field\_t rxsr\_fields[] = {

{"RSVD", 7, 25, 0},

{"PS", 5, 2, 0},

{"OE", 4, 1, 0},

{"BI", 3, 1, 0},

{"PE", 2, 1, 0},

{"FE", 1, 1, 0},

{"DR", 0, 1, 0}

};

unsigned int read\_field\_rxsr(int field\_bit, int field\_size) {

unsigned int rxsr\_value = reg\_rd32(RXSR\_OFFSET);

unsigned int mask =~(~0<<field\_size);

return (rxsr\_value >> field\_bit) & mask;

}

void write\_field\_rxsr(int field\_bit, int field\_size, unsigned int value) {

unsigned int rxsr\_value = reg\_rd32(RXSR\_OFFSET);

unsigned int mask = ((1<< field\_size) - 1) << field\_bit;

rxsr\_value &= ~mask;

rxsr\_value |= (value << field\_bit) & mask;

reg\_wr32(RXSR\_OFFSET, rxsr\_value);

}

void display\_rxsr\_register() {

unsigned int rxsr\_value = reg\_rd32(RXSR\_OFFSET);

printf("RXSR Register Value: 0x%08X\n", rxsr\_value);

printf("Individual fields:\n");

for (int i = 0; i < sizeof(rxsr\_fields) / sizeof(rxsr\_fields[0]); ++i) {

unsigned int field\_value = read\_field\_rxsr(rxsr\_fields[i].start\_bit, rxsr\_fields[i].size\_bits);

printf(" %s: 0x%0\*X\n", rxsr\_fields[i].name, (rxsr\_fields[i].size\_bits + 3) / 4, field\_value);

}

}

int main() {

reg\_setup();

int choice;

unsigned int value;

while (1) {

printf("\nRXSR Register Test Menu\n");

printf("1. Read entire RXSR register\n");

printf("2. Write to entire RXSR register\n");

printf("3. Read specific field\n");

printf("4. Write to specific field\n");

printf("0. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

display\_rxsr\_register();

break;

case 2:

printf("Enter value to write to RXSR register (in hex): ");

scanf("%X", &value);

reg\_wr32(RXSR\_OFFSET, value);

printf("Wrote 0x%X to RXSR register\n", value);

break;

case 3:

printf("Select field to read:\n");

printf("1. RSVD\n");

printf("2. PS\n");

printf("3. OE\n");

printf("4. BI\n");

printf("5. PE\n");

printf("6. FE\n");

printf("7. DR\n");

printf("0. Back to main menu\n");

printf("Enter your choice: ");

scanf("%d", &choice);

if (choice >= 1 && choice <= sizeof(rxsr\_fields) / sizeof(rxsr\_fields[0])) {

unsigned int field\_value = read\_field\_rxsr(rxsr\_fields[choice - 1].start\_bit, rxsr\_fields[choice - 1].size\_bits);

printf("%s Field Value: 0x%X\n", rxsr\_fields[choice - 1].name, field\_value);

} else if (choice != 0) {

printf("Invalid choice.\n");

}

break;

case 4:

printf("Select field to write:\n");

printf("1. RSVD\n");

printf("2. PS\n");

printf("3. OE\n");

printf("4. BI\n");

printf("5. PE\n");

printf("6. FE\n");

printf("7. DR\n");

printf("0. Back to main menu\n");

printf("Enter your choice: ");

scanf("%d", &choice);

if (choice >= 1 && choice <= sizeof(rxsr\_fields) / sizeof(rxsr\_fields[0])) {

printf("Enter value to write (in hex): ");

scanf("%X", &value);

write\_field\_rxsr(rxsr\_fields[choice - 1].start\_bit, rxsr\_fields[choice - 1].size\_bits, value);

rxsr\_fields[choice - 1].stored\_value = value;

printf("Wrote 0x%X to %s field\n", value, rxsr\_fields[choice - 1].name);

} else if (choice != 0) {

printf("Invalid choice.\n");

}

break;

case 0:

printf("Exiting...\n");

return 0;

default:

printf("Invalid choice. Please enter a number between 0 and 4.\n");

break;

}

}

return 0;

}