Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №2

«Конфигурационное пространство PCI»

Выполнил: Проверил:

Студент группы 150504 Преподаватель

Горбачевский К.В. Селезнев А.И.

Минск 2023

1. Постановка задачи

Вывести список всех устройств, подключенных к шине PCI, с их  характеристиками (DevicedID и VendorID состоящие из 4-х символов) в виде таблицы.

1. Краткие теоретические сведения

PCI (Peripheral Components Interconnect) – базовая системная магистраль (шина) компьютера архитектуры x86, предназначенная для подключения внутренних периферийных устройств и контроллеров внешних интерфейсов.

Шина PCI является синхронным параллельным электрическим интерфейсом с общей средой передачи данных (топология «шина»). Состоит из мультиплексированных линий передачи адреса и данных (разделение по времени) и линий различных управляющих сигналов.

Основные характеристики:

• разрядность (ширина) – 32 или 64 бита;

• тактовая частота – 33.3 или 66.6 МГц;

• адресация – 32 или 64 бита (не зависит от ширины шины);

• пропускная способность – от 133 до 533 Мб/с в зависимости от реализации;

• количество подключаемых устройств – зависит от реализации, но не более 32 для одного физического сегмента шины.

В общем случае шина PCI имеет топологию многоуровневая шина: к первичной шине могут подключаться устройства – мосты, управляющие вторичными шинами, и так далее; помимо упомянутых мостов PCI-PCI, к шине подключаются мосты для связи с другими шинами; в их задачи входит трансляция транзакций, поступающих по шине PCI, к устройствам, которые подключены к другой шине.

Транзакция на шине PCI состоит из нескольких фаз. Фактически фаза соответствует одному тактовому интервалу, за время которого по шине передается одна группа данных (32 или 64 бита). В транзакции обязательно имеется одна фаза адресации и одна или несколько фаз передачи данных.

1. Листинг программы

Далее приведен листинг программы, реализующей все поставленные задачи.

#include <stdafx.h>

#include "hexioctrl.h"

#include "pci\_codes.h"

#include <iostream>

#include <conio.h>

#define PCI\_ADDRESS\_PORT 0x0CF8

#define PCI\_DATA\_PORT 0x0CFC

#define PCI\_MAX\_BUSES 128

#define PCI\_MAX\_DEVICES 32

#define PCI\_MAX\_FUNCTIONS 8

#pragma pack(1)

typedef struct \_PCI\_CONFIG\_ADDRESS {

union {

struct {

UINT32 Zero:2;

UINT32 RegisterNumber:6;

UINT32 FunctionNumber:3;

UINT32 DeviceNumber:5;

UINT32 BusNumber:8;

UINT32 Reserved:7;

UINT32 Enable:1;

} s1;

UINT32 Value;

} u1;

} PCI\_CONFIG\_ADDRESS;

#pragma pack()

void printDeviceInfo(ULONG devId, ULONG venId, ULONG baseClass, ULONG subClass, ULONG prog){

std::cout.width(9);

std::cout << std::left << std::hex << devId << "| ";

std::cout.width(9);

std::cout << std::left << std::hex << venId << "| ";

for(int i = 0; i < PCI\_CLASSCODETABLE\_LEN; i++){

if(PciClassCodeTable[i].BaseClass == baseClass &&

PciClassCodeTable[i].SubClass == subClass &&

PciClassCodeTable[i].ProgIf == prog)

{

char result[127];

strcpy(result, PciClassCodeTable[i].BaseDesc);

strcat(result, " ");

strcat(result, PciClassCodeTable[i].SubDesc);

std::cout.width(36);

std::cout << std::left << result << "|";

break;

}

}

std::cout << std::endl;

}

int main(){

PCI\_CONFIG\_ADDRESS cfg;

USHORT bus = 0, dev = 0, func = 0;

ULONG val = 0;

ULONG devId, venId;

ULONG class0, class1, class2, class3, revision;

ULONG subId, subVenId;

int count = 0;

// init

// load driver

ALLOW\_IO\_OPERATIONS;

std::cout<<"DeviceId | VendorId | ClassName |" << std::endl;

std::cout<<"-----------------------------------------------------------" << std::endl;

for(bus = 0; bus < PCI\_MAX\_BUSES; bus++){

for(dev = 0; dev < PCI\_MAX\_DEVICES; dev++){

for(func = 0; func < PCI\_MAX\_FUNCTIONS; func++){

cfg.u1.s1.Enable = 1;

cfg.u1.s1.BusNumber = bus;

cfg.u1.s1.DeviceNumber = dev;

cfg.u1.s1.FunctionNumber = func;

cfg.u1.s1.RegisterNumber = 0;

\_outpd(PCI\_ADDRESS\_PORT, cfg.u1.Value);

val = \_inpd(PCI\_DATA\_PORT);

if (val == 0 || val == -1) continue;

count++;

devId = val >> 16;

venId = val - (devId << 16);

cfg.u1.s1.RegisterNumber = 0x08 >> 2;

\_outpd(PCI\_ADDRESS\_PORT, cfg.u1.Value);

val = \_inpd(PCI\_DATA\_PORT);

class0 = val >> 8;

revision = val - (class0 << 8);

class1 = class0 >> 16;

class3 = class0 - (class1 << 16);

class2 = class3 >> 8;

class3 = class3 - (class2 << 8);

printDeviceInfo(devId, venId, class1, class2, class3);

}

}

}

std::cout<<"-----------------------------------------------------------" << std::endl;

int a;

std::cin >>a;

return 0;

}

1. Тестирование программы

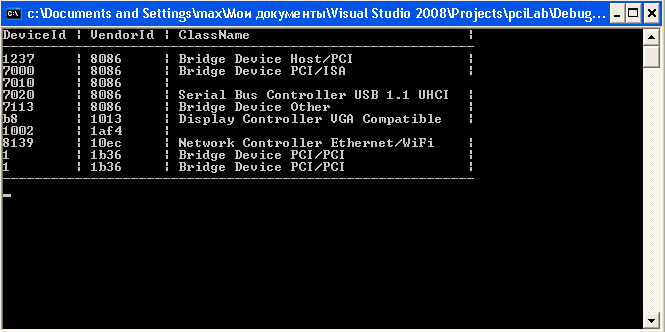


Рисунок 4.1 — Окно вывода программы.

1. Заключение

В данной лабораторной работе была выведена информация о всех устройствах подключенных в данный момент к шине PCI. Все поставленные задачи были выполнены.

Программа написана для операционной системы Windows, использованные инструменты – язык программирования C++, Microsoft Visual Studio.