Міністерство освіти і науки України

Черкаський державний технологічний університет

Кафедра інформаційної безпеки та комп’ютерної інженерії

**З В І Т**

**з лабораторної роботи № 5**

**на тему: «Програмування обміну даними з зовнішніми пристроями»**

**з дисципліни: «Архітектура комп’ютера»**

ПЕРЕВІРИВ: ВИКОНАВ

ст.викладач студент групи КМ-175

Гресько С.О. Косенко Андрій

(оцінка) ( підпис)

( дата, підпис)

ЧЕРКАСИ 2018

**Лабораторна робота № 5**

**Тема: програмування обміну даними з зовнішніми пристроями**

**Мета: вивчення організації системи введення-виведення навчальної моделі ЕОМ**

Технічне забезпечення: персональний комп'ютер, програмна модель навчальної ЕОМ;

Основні поняття: синхронний обмін, асинхронний обмін, програмно керований введення-виведення, введення-виведення по перериванню, векторне переривання.

**Загальні положення**

Зв'язок процесора і ВУ може здійснюватися в синхронному або асинхронному режимі. Синхронний режим використовується для ВУ, завжди готових до обміну. У нашій моделі такими ВУ є дисплей і тоногенератор - процесор може звертатися до цих ВУ, не аналізуючи їх стан (правда дисплей блокує прийом даних після введення 128 символів, формуючи прапор помилки).

Асинхронний обмін передбачає аналіз процесором стану ВУ, яке визначає готовність ВУ видати або прийняти дані або факт здійснення деякої події, контрольованого системою. До таких пристроїв в нашій моделі можна віднести клавіатуру і блок таймерів.

Аналіз стану ВУ може здійснюватися процесором двома способами:

• в програмно-керованому режимі;

• в режимі переривання.

У першому випадку мається на увазі програмне звернення процесора до регістру стану ВУ з подальшим аналізом значення відповідного розряду слова стану. Таке звернення слід передбачити в програмі з певною періодичністю, незалежно від фактичного надходження контрольованого події (наприклад, натискання клавіші).

У другому випадку при виникненні контрольованого події ВУ формує процесору запит на переривання програми, за яким процесор і здійснює зв'язок з ВУ .

**Програмно керований введення-виведення**

Програмно керований введення-виведення, є найбільш простим методом управління, який часто називають також введенням-виведенням з опитуванням. В даному випадку, введення-виведення відбувається під повним контролем центрального процесора (ЦП). Адреса моделі ВУ, до якого здійснюється звернення, вказується в адресній частині команди введення або виведення. ВУ виконує витребувані дію, після чого встановлює в одиницю або нуль відповідний біт в своєму регістрі стану. Нічого іншого, щоб повідомити ЦП, ВУ не робить. Отже, для визначення моменту завершення операції або пересилання чергового елемента блоку даних, процесор повинен періодично опитувати і аналізувати вміст регістрів стану ВУ.

Основний недолік програмно керованого введення-виведення - неефективне використання процесора через очікування готовності чергової порції інформації, протягом якого ніяких інших корисних дій ЦП не виконує. ЦП повинен витрачати час на аналіз бітів стану ВУ, що також негативно позначається на ефективності введення-виведення.

Перевагою ж програмно керованого введення-виведення, є простота, оскільки основні функції з управління введенням-висновком бере на себе процесор. При одночасній роботі декількох ВУ пріоритет пристроїв легко змінити програмними засобами (послідовність опитування)

Введення-виведення по перериваннях

У моделі навчальної ЕОМ передбачений механізм векторних зовнішніх переривань. Зовнішні пристрої формують запити на переривання, що надходять на входи контролера переривань. При підключенні ВУ, здатного формувати запит на переривання, йому ставиться в відповідність номер входу контролера переривань - вектор переривання, що приймає значення в діапазоні 0-9.

Контролер передає вектор, відповідний запит, процесору, який починає процедуру обслуговування переривання.

Кожному з можливих в системі переривань повинен відповідати т. Н. обробник переривання - підпрограма, що викликається при виникненні події конкретного переривання.

Механізм переривань, реалізований в моделі навчальної ЕОМ, підтримує таблицю векторів переривань, яка створюється в оперативній пам'яті моделлю операційної системи (якщо вона використовується) або безпосередньо користувачем.

Номер рядка таблиці відповідає вектору переривання, а елемент таблиці - осередок пам'яті, в трьох молодших розрядах якої розміщується початкова адреса підпрограми, яка обслуговує переривання з цим вектором.

Таблиця переривань в розглянутій моделі жорстко фіксована - вона займає осередку пам'яті з адресами 100-109. Таким чином, адреса обробника з вектором 0 повинен розташовуватися в осередку 100, з вектором 2 - в осередку 102. При роботі з перериваннями не рекомендується використовувати осередки 100-109 для інших цілей.

Процесор починає обробку переривання (якщо вони дозволені), завершивши поточну команду. При цьому він:

1. Отримує від контролера вектор переривання.

2. Формує і поміщає в верхівку стека слово, три молодших розряду ([3: 5]) якого - поточне значення PC (адреса повернення з переривання), а розряди [1: 2] зберігають десятковий еквівалент шестнадцатеричной цифри, що визначає значення вектора прапорів ( I, OV, S, Z). Наприклад, якщо I = 1, OV = 0, S = 1, Z = 1, то в розряди [1: 2] запишеться число 1110 = 10112.

3. Скидає в 0 прапор дозволу переривання I.

4. Витягує з таблиці векторів переривань адреса обробника, відповідний обслуживаемому вектору, і поміщає його в PC, здійснюючи тим самим перехід на підпрограму обробника переривання.

Таким чином, виклик обробника переривання, на відміну від виклику підпрограми, пов'язаний з приміщенням в стек не тільки адреси повернення, а й поточного значення вектора прапорів. Тому останньою командою підпрограми обробника повинна бути команда IRET, яка не тільки повертає в PC три молодші розряду осередки - верхівки стека (як RET), а й відновлює ті значення прапорів, які були в момент переходу на обробник переривання.

Не всяке подія, яке може викликати переривання, призводить до переривання поточної програми. До складу процесора входить програмно-доступний прапор I дозволу переривання. При I = 0 процесор не реагує на запити переривань. Після скидання процесора прапор I так же скинуто і все переривання заборонені. Для того щоб дозволити переривання, слід в програмі виконати команду EI (від англ. Enable interrupt). Вище зазначалося, що при переході на обробник переривання прапор 1 автоматично скидається, в цьому випадку перервати обслуговування одного переривання іншим перериванням не можна. За командою IRET значення прапорів відновлюється, в т. Ч. Знову встановлюється I = 1, отже, в основній програмі переривання знову дозволені.

Якщо потрібно дозволити інші переривання в обробнику переривання, досить в ньому виконати команду EI. Контролер переривань і процесор на апаратному рівні блокують спроби запустити переривання, якщо його обробник почав, але не завершив роботу.

Таким чином, прапор 1 дозволяє або забороняє всі переривання системи. Якщо потрібно вибірково дозволити деяку підмножину переривань, використовуються програмно-доступні прапори дозволу переривань безпосередньо на зовнішніх пристроях.

Як правило, кожне зовнішнє пристрій, який може викликати переривання, містить в складі своїх регістрів розряд прапора дозволу переривання (див. Формат регістрів CR і CTR на рис. 2.41, 2.45), за замовчуванням встановлений в 0. Якщо залишити цей прапор в нулі, то зовнішнього пристрою забороняється формувати запит контролеру переривань.

Іноді буває зручно (наприклад, в режимі налагодження) мати можливість викликати оброблювач переривання безпосередньо з програми. Якщо використовувати для цих цілей команду CALL, яка поміщає в стек тільки адреса повернення, то команда IRET, розміщена останньої в обробнику, може спотворити значення прапорів (всі вони будуть скинуті в 0, т. К. Команда CALL формує тільки три молодші розряду осередку верхівки стека, залишаючи інші розряди в 000).

Тому в системах команд багатьох ЕОМ, в т. Ч. І нашої моделі, є команди виклику переривань - INT n (в нашій моделі n Є {0, 1,9}), де n - вектор переривання. Процесор, виконуючи команду INT n, виробляє ті ж дії, що і при обробці переривання з вектором n.

Характерно, що за допомогою команди INT n можна викликати оброблювач переривання навіть в тому випадку, коли прапор дозволу переривання I скинутий.

**Порядок виконання роботи**

1) Запустити програмну модель навчальної ЕОМ;

2) Підключити до неї певні в завданні зовнішні пристрої. Варіанти завдання вказані в таблиці 2.29.

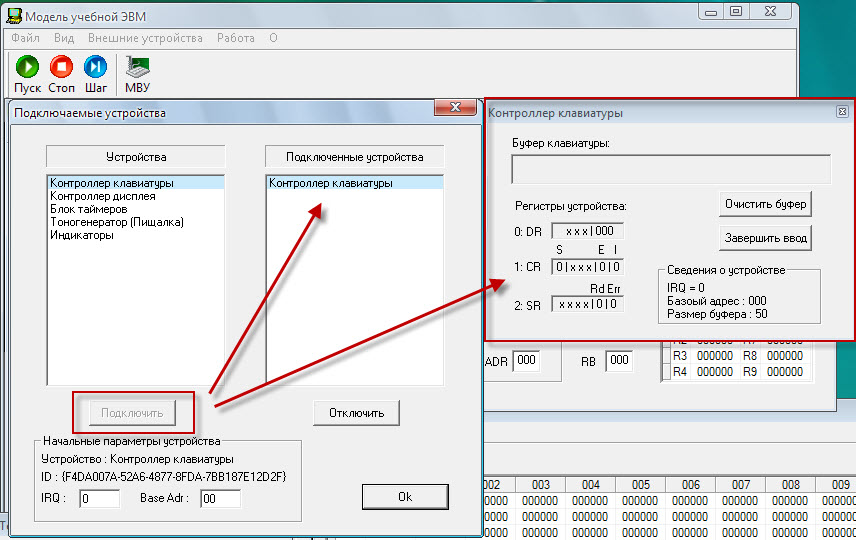
Відкрийте меню Зовнішні пристрої, далі виберіть Менеджер ВУ. З'явиться вікно «Пристрої, що підключаються».

Вікно розділене на дві частини: Пристрої - в цій частині представлені зовнішні пристрої підтримувані в навчальній моделі ЕОМ; Підключення пристрою - в цій частині, ті зовнішні пристрої, які підключені вами. Щоб підключити будь-який пристрій, у вікні **Пристрої** потрібно вибрати необхідне зовнішнє пристрій і натиснути кнопку **Підключити**, при цьому з'явиться вікно оглядача зовнішнього пристрою.Вікно розділене на дві частини: Пристрої - в цій частині представлені ВУ підтримувані в навчальній моделі ЕОМ; Підключення пристрою - в цій частині, ті пристрої ВУ, які підключені вами. Щоб підключити будь-який пристрій, у вікні Пристрої потрібно вибрати необхідне ВУ.

Наприклад, для виконання завдання потрібно контролер клавіатури, для цього слід виконати наступні дії:

• лівою кнопкою миші вибрати контролер клавіатури, малюнок 2.36;

• далі, натиснути кнопку Підключити. При цьому, відобразиться вікно оглядача Контролер клавіатури і вибране вами ВУ з'явиться в правій частині вікна Підключені пристрої, малюнок 2.37.

Аналогічно підключаться інші зовнішні пристрої. Більш докладно з описом оглядачів ВУ можна ознайомитися в розділі 2.8. Підключення ВУ, також відображаються в меню

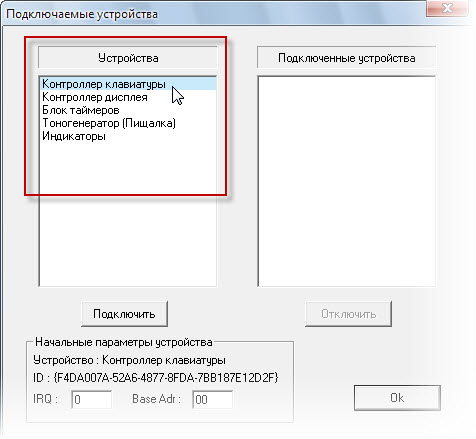
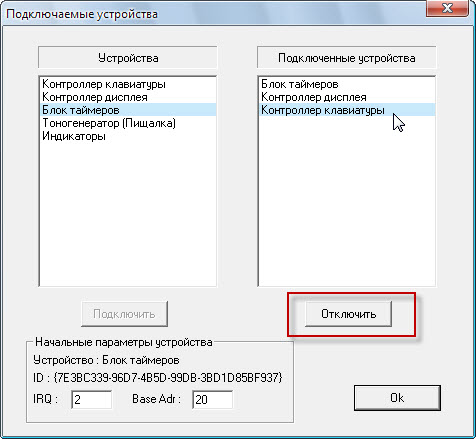
Зовнішні пристрої, головного вікна навчальної ЕОМ.Рисунок 2.36 – Вибір пристрою

Рисунок 2.37 – Підключення контролера клавіатури

Щоб відключити ВУ, необхідно виконати зворотні дії: мишею вибрати відключається ВУ і натиснути кнопку Відключити, малюнок 2.38

Малюнок 2.38

Відключення зовнішнього пристрою.

3) Скласти і налагодити програму, передбачену завданням, з використанням програмного аналізу прапорів готовності ВУ.

**Розглянемо приклади програмування ВУ.**

Контролер клавіатури. Контролер клавіатури являє собою модель зовнішнього пристрою, що приймає ASCII-коди (таблиця 2.23) від клавіатури ПЕОМ. Символи поміщаються послідовно в буфер символів, розмір якого встановлений рівним 50 символам, і відображаються у вікні оглядача. Приклад програми посимвольного введення наведено в таблиці 2. 22.

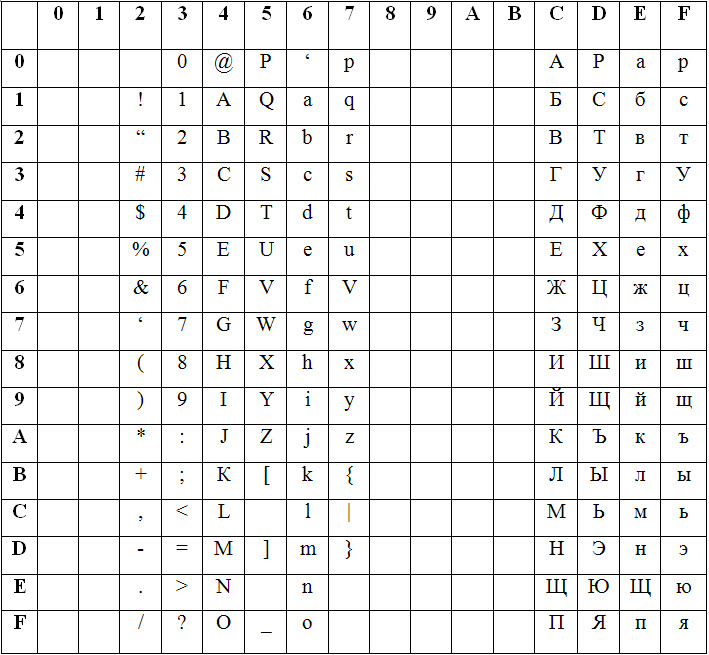
Контролер клавіатури виконує чотири команди при виведенні відповідних кодів за адресою 1, таблиця 2.21 [2].Таблица 2.21 – Коды команд контроллера клавиатуры

|  |  |
| --- | --- |
| **Десятичное число** | **Команда** |
| 101 | Очистить буфер (эквивалентно кнопке Очистить буфер) |
| 102 | Сбросить флаг Err в регистре SR |
| 103 | Установить в «1» флаг S в регистре CR |
| 104 | Сбросить в «0» флаг S в регистре CR |

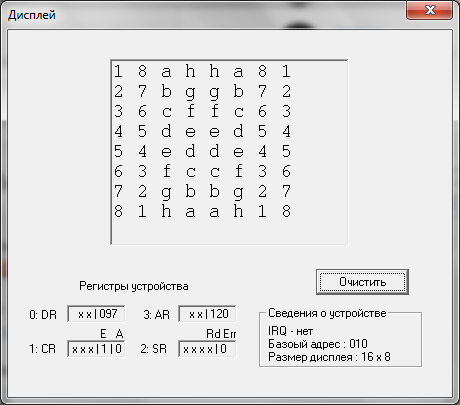
Таблица 2.22 – Программа посимвольного ввода с клавиатуры

|  |  |  |
| --- | --- | --- |
| **Адрес** | **Мнемокод** | **Примечания** |
| 000 | RD #10 | Установить флаг Е в регистре CR |
| 001 | OUT 1 | Включить клавиатуру |
| 002 | OUT 11 | Включить дисплей |
| 003 | RD #103 | Передать в контроллер код команды |
| 004 | OUT 1 | Установить S в «1» (режим посимвольного ввода) |
| 005 | M1: IN 2 | Проверка нажатия – флага готовности Rdy |
| 006 | JZ M1 | Ожидание Rdy = 1 |
| 007 | IN 0 | Считывание введенного символа из буфера в аккумулятор |
| 008 | OUT 10 | Передача ASCII-кода (например) на символьный дисплей |
| 009 | JMP M1 | Возврат к ожиданию следующего нажатия |

Таблица 2.23 – Таблица кодов ASCII, фрагмент



**Результат виконаної роботи**



**Блок таймерів**. Блок таймерів включає три однотипних каналу таймерів Т1 (адреса 1), Т2 (адреса 3), Т3 (адреса 5) зі своїми регістрами управління CT1R (адреса 2), CT2R (адреса 4), CT3R (адреса 6) і загальний регістр станів SR (адреса 0).

Скидання прапорів переповнення таймерів в регістрі SR здійснюється тільки програмно по командам, представленим в таблиці 2.27. Константи кодів цих команд повинні виводиться за адресою 0 блоку таймерів (регістра SR).Таблиця 2.27 – Код команди таймера

|  |  |
| --- | --- |
| **Десятичное число** | **Команда** |
| 101 | Сбросить флаг переполнения Т1 |
| 102 | Сбросить флаг переполнения Т2 |
| 103 | Сбросить флаг переполнения Т3 |

У таблиці 2.28 представлений приклад програми, за допомогою якої реалізується подача звукового сигналу кожні 10 сек.

Такт таймера може змінюватися в залежності від ЕОМ, на якій працює модель. Якщо вважати, що тривалість такту становить 16 мс, то для відліку відрізка часу в 10 сек. буде потрібно 10000/16 = 625 тактів.

Таблиця 2.28 - Текст програми використання тоногенератора

|  |  |  |
| --- | --- | --- |
| **Адрес** | **Мнемокод** | **Примечания** |
| 000 | M0: RD #625 | Загрузка константы, соответствующей задержке в 10 сек |
| 001 | OUT 21 | в регистр таймера Т1 |
| 002 | RDI 1101 | Запуск таймера с коэффициентом деления К=1 |
| 004 | OUT 22 | на вычитание, прерывание запрещено |
| 005 | M1: IN 20 | Анализ регистра состояния блока таймеров |
| 006 | JZ M1 | и возврат, если флаг переполнения не установлен |
| 007 | RDI 1000 | и останов |
| 009 | OUT 22 | таймера |
| 010 | RD #101 | Сброс |
| 011 | OUT 20 | флага переполнения FT1 |
| 012 | RD #200 | Задать частоту звучания 200 Гц |
| 013 | OUT 30 | в регистр частоты тоногенератора |
| 014 | RD #300 | Задать длительность звучания – 300 мс |
| 015 | OUT 31 | и включить звук |
| 016 | JMP M0 | Повторить бесконечный цикл |

**Результат виконаної роботи**