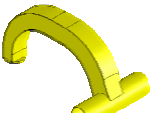
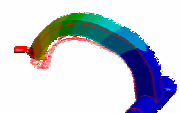
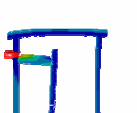
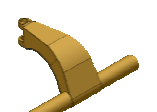
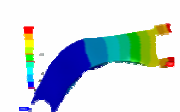
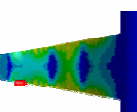

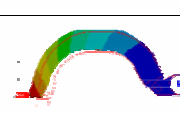
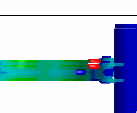
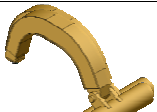
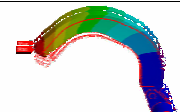
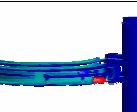
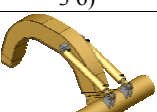
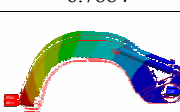
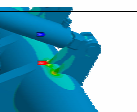
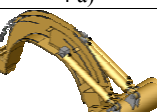
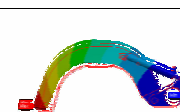
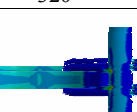
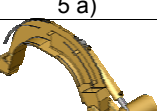
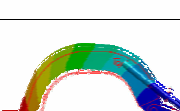
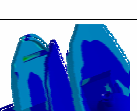


Таблиця 1 – Результати порівняльного аналізу напруженого стану арок тягових рам.

Моделі арок	Маса кг	Деформація мм	Напруження МПа
 1	410.18	 7.5476	 322.12 (переріз)
 2	728.49	 6.304	 340.845
 3 а)	672.73	 6.6482	 200.07
 3 б)	672.73	 6.7884	 199.77
 4 а)	750.63	 8.4339	 320
 5 а)	692.08	 8.2318	 259.45
 5 б)		 9.6774	 343

1. Рекомендувати використовувати раму конструкції 5, в якій арка приєднана до тягової рами за допомогою шарніра, а її положення відносно самої рами регулюється за допомогою двох гідроциліндрів.

2. Товщину середньої і бокової стінок листів рекомендовано зменшити з 12мм до 10мм.

3. У місцях криволінійного вигину перетину додатково встановити середню стінку та кронштейни, які знижують напруження в цих зонах (рис. 2).

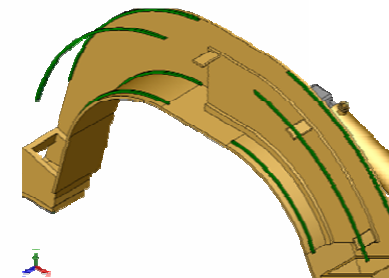


Рисунок 2 – Схема розміщення додаткової середньої стінки та кронштейнів

### Список літератури

1. Холодов А.М. Проектирование машин для земляных работ / А.М. Холодов. – Х.: Вища школа, 1986. – 272 с.
2. Алямовский А.А. SolidWorks/COSMOSWorks. Инженерный анализ методом конечных элементов / А.А. Алямовский. – М.: ДМК Пресс, 2004. – 432 с.
3. Перельмутер А.В. Расчетные модели сооружений и возможность их анализа / А.В. Перельмутер, В.И. Сливкер. - М.: ДМК Пресс, 2007. – 600 с.

### ПРОГРАМНА РЕАЛІЗАЦІЯ НАВЧАЛЬНОГО ЕМУЛЯТОРА КОНТРОЛЕРА ПЕРЕРИВАНЬ

М.О. Баранець, ст. гр. КІ 09-3, О.П. Дібреньський, викладач  
Кіровоградський національний технічний університет

Становлення високоефективної світової економіки неможливе без розвинуеного ринку комп’ютерних засобів та програмного забезпечення. На сучасному етапі розвитку технологій комп’ютери є найбільш необхідними та високопродуктивними машинами, тому у вищих навчальних закладах України вже з’явилося багато спеціальностей, що здійснюють підготовку програмістів.

Дослідження [1-6] показали, що в процесі підготовки ІТ-фахівців виникає нагальна проблема у сучасних навчальних засобах, серед яких програмні імітаційні моделі (емулятори) пристроїв та систем, мов програмування тощо. Але не всі з перерахованих ІТ-інструментів може забезпечити вітчизняний ринок програмного забезпечення. Отже, постає нагальна потреба в у їх розробці, що є актуальним завданням.

Метою роботи є розроблення навчальної імітаційної моделі (емулятора) мікросхеми контролера переривань (КП), з метою її впровадження у навчальний процес підготовки фахівців за напрямом «Комп'ютерна інженерія» кафедри програмного забезпечення Кіровоградського національного технічного університету.

Структурна й функціональна схеми, принципи роботи емулятора роботи контролера переривань повинні строго відповідати [7]. Головне вікно розробленого програмного забезпечення КП наведено на рисунку 1.

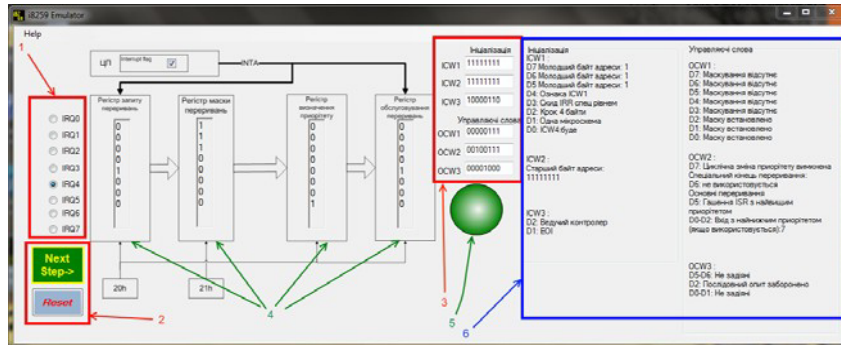


Рисунок 1 – Головне вікно програми-імітатора контролера переривань

Головне вікно програми-імітатора (рисунок 1) умовно поділене на три частини:

- демонстрація роботи контролера і засоби управління демонстрацією;
- введення управляючих слів («програмування») КП;
- розшифрування значущих бітів управляючих слів (для зручності користувача).

На початку роботи з розробленим емулятором усі елементи управління демонстрацією заблоковані. Після введення кожного наступного управляючого слова проводиться його аналіз на наявність помилок введення (неправильні біти-ознаки, конфлікт режимів, недопустима комбінація бітів), а також перевірка порядку вводу слова. Якщо одна з перевірок дає невірний результат, зелена «лампочка» внизу блока змінює колір на червоний. При успішному проходженні перевірок, дозволяється ввести наступне управляюче слово, а у відповідному віконці справа з'являється розшифровка значення бітів введенного слова.

По закінченні «програмування» елементи управління демонстрацією активуються, а у віконцях регістрів з'являються відповідні значення. Це сигналізує про готовність емулятора до демонстрації, після чого користувач обирає вхід, на який подається запит на переривання (за замовчанням – IRQ0). Запуск демонстрації обробки переривання здійснюється кнопкою «Next step». З кожним натисненням відображається наступний етап обробки сигналу.

Серед усіх функцій, що реалізовані у програмі найголовнішою є функція intr(int num), котра відповідає власне за емуляцію процесу роботи контролера переривань та його візуалізацію. Аргумент функції – номер обраного користувачем входу переривань. Далі за допомогою оператора вибору switch функція визначає, скільки разів була натиснута кнопка «Next». В залежності від номеру поточного натиснення вибирається дія, яку потрібно виконати. Всього для повної демонстрації потрібно сім кліків, а саме:

0 – визначення входу переривань, на який було подано запит; встановлення біту IRR, котрий відповідає цьому входові і вивід його на екран;

1 – перехід до маскового регістру: перевірка того, чи не замаскований обраний користувачем вхід переривань; якщо вхід замаскований, то емулятор скидається у вихідний стан;

2 – перехід до регістру визначення пріоритету: якщо увімкнена ротація пріоритету, то значення цього регістру встановлюється "так". Для наступних запитів поточний запит матиме найнижчий пріоритет;

3 – візуалізація виставлення центральному процесору сигналу INT;

4 – якщо не встановлено Interrupt Flag, то емулятор скидається у вихідний стан; якщо ж встановлено, то емулюється відповідь процесора сигналом INTA; перевіряється чи потрібно скидати регістр IRR по встановленню ISR (вказано в управляючому слові ICW1, біт D3), якщо так, то регістр скидається;

5 – візуалізація кінця переривання. Якщо встановлений автоматичний кінець переривання, то регістр ISR скидається відразу, а якщо ж встановлений звичайний кінець переривання, то виводиться повідомлення про надсилання у порт 20h значення 20h, після чого ISR скидається.

7 – завершення демонстрації. Значення «б» пропущено навмисно, щоб створити певну паузу між кінцем роботи контролера та завершенням демонстрації. По сьомому кліку область демонстрації повертається у стан готовності до демонстрації. Наступний клік кнопки «Next» почне імітацію роботи КП спочатку.

Для роботи з програмою-імітатором необхідно ввести усі керуючі слова у блоці «програмування» (3) на рисунку 1 (тут і надалі всі елементи, позначені на рисунку 1 позначатимуться цифрою у круглих дужках). Після введення кожного управляючого слова в області розшифровки значень (6) з'являється пояснення щодо поточного значення бітів у введеному керуючому слові. Після введення останнього управляючого слова активуються елементи управління демонстрацією (2), вибору входу переривання (1) та область демонстрації (4). Програма готова до емуляції. Наступний етап роботи – вибір одного з входів запиту переривання для подальшої емуляції. Вибір здійснюється шляхом натиснення на обраний канал в області (1). Наступний крок – емуляція обробки запиту переривання. Запуск кожного наступного кроку виконується натисненням на кнопку «Next step». Також емулятор забезпечений функцією скиду (кнопка «Reset»).

Слід виділити особливості розробленого ПЗ-імітатора КП:

– управляючі слова вводяться у строгому порядку ICW1→ICW2→ICW3→OCW1→OCW2→OCW3; при спробі ввести їх у іншому порядку вмикається індикатор (5), поле введення очищується і для повторного його введення потрібно ввести, або змінити попередні управляючі слова у вищевказаному порядку.

– при допущенні помилки у введенні управляючого слова емулятор виконує дії аналогічні тим, що проводяться у випадку введення управляючих слів у неправильному порядку.

У запропонованій програмі-емюляторі реалізовано потужну систему допомоги, підказок, а також інструкцію користувача.

Отже, у роботі запропоновано і обґрунтовано програмну реалізацію емулятора (імітатора) контролера переривань з метою впровадження розробленого програмного засобу у навчальний процес підготовки фахівців з програмування ВІЗ.

## Список літератури

1. Сидоренко В.В. Використання програмних емуляторів пристроїв обчислювальної техніки в навчальному процесі. / В.В. Сидоренко, О.В. Коваленко, О.П. Доренський. // Матеріали III Міжнародної науково-технічної конференції “Комп'ютерні системи та мережні технології” (CSNT-2010). – К.: НАУ, 2010. – С. 89.

- Дима Я.Ю. Емулятори вимірювальних приладів як інструмент ІКТ для реалізації міжпредметних зв'язків фізики та математики. / Я.Ю. Дима, І.В. Лапенко, О.В. Сасенко. // Збірник наукових праць Бердянського державного педагогічного університету. – 2011. – № 3. – С. 77-83.
- Завгородній О.К. Програмне забезпечення візуалізації процесу роботи і системного програмування передачі даних через USB. // Збірник тез доповідей XLIII наукової конференції студентів і магістрів Кіровоградського національного технічного університету. – Кіровоград: РВЛ КНТУ, 2009. – С. 65
- Валова К.М. Програма-емулятор системного таймера. // Збірник тез доповідей XLI наукової конференції студентів і магістрів Кіровоградського національного технічного університету. – Кіровоград: РВЛ КНТУ, 2007. – С. 65.
- Мельниченко Т.М. Програмний емулятор системного та реального часу на платі IBM PC в навчальному процесі підготовки системних програмістів. // Матеріали III Міжнародної науково-технічної конференції “Комп'ютерні системи та мережні технології” (CSNT-2010). – К.: НАУ, 2010. – С. 73.
- Гуртовий Є.О. Програма-емулятор годинника реального часу (RTC) на платі IBM PC. // Збірник тез доповідей XL наукової конференції студентів і магістрів Кіровоградського національного технічного університету. – Кіровоград: РВЛ КНТУ, 2006. – С. 66.
- Гуржій А.М. Архітектура, принципи функціонування та керування ресурсами IBM PC: Навчальний посібник. / А.М. Гуржій, С.Ф. Коряк, В.В. Самсонов, О.Я. Скларов. – Х.: “Компанія СМІТ”, 2003. – 511 с.

## АНАЛІЗ ВИМОГ ДО ІНТЕРФЕЙСУ КОРИСТУВАЧА КОМП'ЮТЕРНИХ ІМІТАЦІЙНИХ МОДЕЛЕЙ ПРИСТРОЮ (СИСТЕМИ)

**А.О. Максимов**, ст. гр. КІ 09-3, **О.П. Дібреньський**, викладач  
Кіровоградський національний технічний університет

Впровадження інформаційних освітніх технологій у навчальних закладах є одним з головних чинників у підготовці високоякісного фахівця [1-3], чільне місце серед яких посідають широко застосовані у ВНЗ програмні імітаційні моделі (емулятори) пристроїв [2-3], процесів, явищ тощо.

Дослідження [2-5] показало, що на сьогоднішній день застосування у навчальному процесі навчальних закладів (НЗ) програмних імітаційних моделей (емуляторів) пристроїв, мікросхем, систем та процесів є перспективним, економічно вигідним та ефективним для організації якісного навчального процесу. Вони також є особливо важливими для створення гнучкої і сучасної лабораторної бази НЗ з врахуванням стрімкого розвитку обчислювальної техніки.

Метою роботи є аналіз вимог до інтерфейсу користувача комп'ютерної імітаційної моделі пристрою (системи) та підбиття інформації про створення більш відповідний інтерфейс для будь-якого ПЗ.

Користувач завжди повинен відчувати, що саме він керує програмним забезпеченням, а не навпаки. Для створення у користувача такого відчуття „внутрішньої свободи” інтерфейс повинен мати такі властивості: природність, узгодженість, дружність, принцип “зворотного зв'язку”, простота, гнучкість, естетична привабливість.

Ефективність інтерфейсу полягає у швидкому розвитку в користувачів простої концептуальної моделі взаємодії. Це досягається через узгодженість. Концепція узгодженості полягає в тому, що під час роботи з ПК у користувача формується система очікування однакових реакцій на однакові дії, що постійно підкріплює призначену для користувача модель інтерфейсу.

Інтерфейс може бути узгоджений в трьох аспектах: фізичному, синтаксичному і семантичному. Фізична узгодженість відноситься до технічних засобів: схема клавіатури, розташування клавіш, використання миші. Наприклад, для клавіші F3 фізична узгодженість має місце, якщо вона завжди знаходиться в одному і тому ж місці, незалежно від обчислювальної системи. Синтаксична узгодженість відноситься до послідовності й порядку появи елементів на екрані (мова зображень), послідовності запитів (мова дій). Наприклад, синтаксична узгодженість матиме місце, якщо заголовок панелі завжди розміщується в центрі і вгорі панелі. Семантична узгодженість відноситься до значення елементів, складових інтерфейсу.

Природність інтерфейсу – найважливіша його характеристика, оскільки він не змушує користувача істотно змінювати звичні для нього способи розв'язання задачі. Це, зокрема, означає, що повідомлення і результати, які видаються програмним продуктом, не повинні вимагати додаткових пояснень.

Використання знайомих користувачеві понять і образів (метафор) забезпечує інтуїтивно зрозумілий інтерфейс при виконанні завдань. Наприклад, на відміну від свого паперового аналога, папка на робочому столі ОС Windows може використовуватися для зберігання цілого ряду інших об'єктів (таких, наприклад, як принтери, калькулятори тощо).

Завжди забезпечуйте зворотний зв'язок для дій користувача. Кожна дія користувача повинна отримувати візуальне, а іноді й звукове підтвердження того, що програмне забезпечення сприйняло введену команду; при цьому вид реакції, за можливістю, повинен враховувати природу виконаної дії.

Зворотний зв'язок ефективний в тому випадку, якщо він реалізується своєчасно, тобто якомога ближче до точки останньої взаємодії користувача з системою. Коли комп'ютер обробляє завдання, що надійшло, корисно надати користувачеві інформацію щодо стану процесу, а також можливість перервати цей процес у разі потреби. Ніщо так не бентежить не дуже досвідченого користувача, як заблокований екран, який ніяк не реагує на його дії. Типовий користувач здатний витерпіти лише декілька секунд очікування реакції від свого електронного “співбесідника”.

Інтерфейс повинен бути простим. При цьому мають на увазі не спрощеність, а забезпечення легкості в його вивченні і у використанні. Крім того, він повинен надавати доступ до всього набору функціональних можливостей, передбачених даною програмою. Реалізація доступу до широких функціональних можливостей і забезпечення простоти роботи суперечать один одному.

Один з можливих шляхів підтримки простоти – зображення на екрані інформації, мінімально необхідної для виконання користувачем чергового кроку завдання.

Інший шлях до створення простого, але ефективного інтерфейсу – розміщення та представлення елементів на екрані з урахуванням їх смислового значення й логічного взаємозв'язку.

Під гнучкістю інтерфейсу розуміють здатність враховувати рівень підготовки і продуктивність праці користувача. Властивість гнучкості припускає можливість зміни структури діалогу або вхідних даних.

Проектування візуальних компонентів є найважливішою складовою розробки програмного інтерфейсу. Коректне візуальне представлення об'єктів забезпечує передачу дуже важливої додаткової інформації про поведінку та взаємодію різних об'єктів. В той же час слід пам'ятати, що кожен візуальний елемент, який з'являється на екрані, потенційно вимагає уваги користувача, яка, як відомо, не безмежна. Слід забезпечити формування на екрані такого середовища, яке не лише сприяло б розумінню користувачем представленої інформації, але й дозволяло б зосередитися на найбільш важливих її аспектах. До візуальних атрибутів інформації, що відображається, відносяться: взамне розташування та розмір об'єктів, що відображаються; палітра; засоби привертання уваги користувача.