

УДК 681.3.06

**В.А. Хорошко,**  
доктор технических наук, профессор,  
**Ю.Е. Хохлачева,**  
кандидат технических наук, доцент,  
**Л.И. Моржова**

## ВЫБОР ЭЛЕМЕНТНОЙ БАЗЫ ДЛЯ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА

*В работе рассматривается методика выбора элементной базы для бортового вычислителя беспилотного летательного аппарата и его проектирование.*

**Ключевые слова:** элементная база, микропроцессорные наборы, бортовой вычислитель, беспилотный летательный аппарат.

*В роботі розглядається методика вибору елементної бази для бортового обчислювача безпілотного літального апарату і його проектування.*

**Ключові слова:** елементна база, мікропроцесорні набори, бортовий обчислювач, безпілотний літальний апарат.

*In this paper the method of choice for the elemental base for board calculator of unmanned aerial vehicle and its design is considered.*

**Keywords:** electronic components, chipset, onboard computer, unmanned aerial vehicle.

### Введение

Выбор элементной базы и проектирования бортового вычислителя (далее – БВ) для беспилотного летательного аппарата (далее – БПЛА) весьма сложная и ответственная задача, при решении которой необходимо учитывать различные факторы.

Большой выбор микропроцессорных наборов (далее – МПН) и однокристалльных микро-ЭВМ (далее – ОМЭВМ) требует системного подхода к выбору аппаратного обеспечения БВ.

Так как в настоящее время БПЛА являются основными элементами информационно-разведывательными средствами обеспечения информацией, поэтому резко возросли требования к их бортовой аппаратуре.

Важнейшими аспектами эффективности применения БПЛА является:

- 1) развитие и усовершенствование систем видеонаблюдения обстановки;
- 2) разработка БВ;
- 3) предоставление информационного обеспечения, достаточного для принятия решений;
- 4) увеличение разведзащищенности каналов связи и управления БПЛА;
- 5) увеличение скорости передачи данных в каналах связи;
- 6) разработка и внедрение новых концепций, реализация которых является принятой по критерию “эффективность-стоимость-время” [1].

Как видно из выше сказанного все пункты завязаны на пункте “2”. Кроме того, цифровое оборудование БПЛА все больше приближается по своим характеристикам к интеллектуальным микропроцессорным системам, что позволяет создавать очень гибкую политику проведения разведки, приближающуюся по

своим функциям и механизмам перестраиваемых вычислительных структур. Именно поэтому современные БВ можно называть составной частью интеллектуальной разведывательной системы (далее – ИРС), реализованной на базе МПН и ОМЭВМ [2].

Потребность в решении сложных задач обработки информации в реальном масштабе времени с высокой точностью определяет актуальность проблемы выбора элементной базы для БВ. При этом сложная задача представляется в виде совокупности взаимосвязанных подзадач, которые решаются последовательно-параллельно.

#### **Анализ публикации**

В настоящее время используются несколько методов выбора МПН и ОМЭВМ. Один из методов использует бенчмарковские программы, ускоряет процесс разработки за счет того, что длина бенчмарка ограничена и содержит 100-200 команд, анализируемых МПН и ОМЭВМ, однако при их применении при составлении программы требует изучения и учета систем команд всех МПН и ОМЭВМ, что достаточно сложно и трудоемко [3; 4].

Другой подход к выбору заключается в следующем. Разработчику необходимо составить таблицу основных характеристик всех существующих МПН и ОМЭВМ, ограничиваясь выбором из нескольких характеристик одинаковой значимости (не более 12) [4; 5]. Эталонным считается гипотетический микропроцессор (а для ОМЭВМ однокристалльная ЭВМ), с лучшими показателями. Далее они сравниваются по всем показателям с базовыми. Микропроцессор или микро ЭВМ, характеристики которого наиболее близки к эталонным, считается лучшим. Оценки, полученные при использовании данной методики, не отражают многих особенностей и специфики их применения на объекте, поэтому являются приближенными и пригодны только для предварительной ориентации. Таким образом, сведения различные по своей природе показателей в единую целевую функцию с параметрами, определяемыми с помощью экспертных оценок, трудоемко из-за большого объема вычислений. Кроме того, данная методика не ориентирована на проектирование и разработку БВ и не учитывает структуру алгоритма, для которого разрабатывается БВ.

Наиболее перспективным методом, который используют при разработке БВ, являются кросс-моделирующие средства [6]. Однако в этом случае разработчику необходимо обращаться к банку данных кросс-моделирующих программ, ориентированных на построение и отладку программ для всех микропроцессоров и ОМЭВМ на языке высокого уровня, что не везде приемлемо.

#### **Цель работы**

В статье предлагается автоматизированная процедура выбора компонентов для проектируемых БВ по критерию быстроедействие – аппаратные затраты.

#### **Основная часть**

Алгоритм процедуры выбора элементной базы для БВ приведен на рис.1. Суть метода заключается в следующем. Разработке по заданному алгоритму, который должен быть реализован в БВ, составляет программу на языке высокого уровня, которая реализуется на ЭВМ. В процессе ее выполнения составляется таблица, где дается информация о количестве содержащихся в объектном модуле коротких машинных операций типа: сложения, вычитания, пересылки и сдвига. По завершении выполнения исходной программы анализирующая программа обращается по второй таблице, в которой приведены времена выполнения коротких операций различными МПН и ОМЭВМ. Перемножив соответствующие графы двух таблиц, а затем просуммировав результаты, рассчитывают предварительную оценку времени выполнения заданного алгоритма на всех МПН или ОМЭВМ.

Даже из множества всех МПН или ОМЭВМ с помощью оценок выбирают такие, которые удовлетворяют по быстроедействию техническое задание (далее –

ТЗ). В полученном подмножестве находим элемент, требующий минимальных затрат по основным критериям, заданным в ТЗ, к которым относятся: объем памяти запоминающего устройства  $L$ ; разрядность чисел  $n_ч$ , команд  $n_к$ , запоминающего арифметического устройства  $n_{зв}$ ; скорость ввода  $V_{вв}$  и вывода  $V_{выв}$  информации. Затем разрабатывается программа в системе команд выбранного микропроцессора или ОМЭВМ и рассчитываются точные характеристики по критериям быстродействие – аппаратные затраты (т.е. быстродействие  $V$ ). Большое внимание в БВ уделяется требованиям надежности, безотказной работе  $P(T)$  и готовности вычислителя  $K_r(t)$  к работе. Так же, к основным характеристикам БВ относят: потребляемую мощность, габариты, массу и требования эксплуатации (допустимые значения постоянных перегрузок, амплитуды и частоты вибрации, давления и температуры окружающей среды и т.п.). Кроме того, обязательно указывается режим работы БВ по времени.

Быстродействие БВ определяется при помощи следующего выражения:

$$V = \frac{1}{\sum_{i=1}^n n_i f_i T_i}, \quad (1)$$

где  $f_i$  – частота выполнения операции  $i$ -го типа,  
 $T_i$  – время на выполнение этой операции.

Разрядность чисел для БВ с плавающим запятой в двоичных разрядах.

$$n_2 = n_{3п} + n_{п} + n_{3м} + n_{м}, \quad (2)$$

где,  $n_{3п}$  – разрядность знака порядка;

$n_{п}$  – разрядность порядка;

$n_{3м}$ ,  $n_{м}$  – разрядность соответственно знака мантиссы и самой мантиссы.

Разрядность чисел для БВ с фиксированной запятой двоичных разрядах

$$n_r' = n_{3п} + n_{м}. \quad (3)$$

Разрядность запоминающего устройства определяется длиной слова, которое может быть записано в ячейку памяти (т.е. максимальная длина слова (числа), которое может перерабатываться).

Объем памяти запоминающего устройства

$$L = \mu + n, \quad (4)$$

где  $\mu$  – количество ячеек запоминающего устройства;

$n$  – разрядность одной ячейки.

Скорость работы БВ на ввод и на вывод описывается следующим образом:

$$\left\{ \begin{array}{l} V_{\text{вв}} = \frac{1}{\tau_{\text{вв}}} \\ V_{\text{выв}} = \frac{1}{\tau_{\text{выв}}} \end{array} \right., \quad (5)$$

где  $\tau_{\text{вв}}$  и  $\tau_{\text{выв}}$  – время, приходящееся соответственно на ввод и вывод одного  $n$ -разрядного числа.

Для оценки надежности БВ выбираем характеристику  $W$ , определяющую вероятность правильности вычислений:

$$W = k_r(t)P(T). \quad (6)$$

где  $T$  – время выполнения задачи;  
 $k_r$  – коэффициент готовности БВ [6].

Коэффициент готовности определяется по времени наработки на отказ  $T_{но}$  и среднему времени для устранения неисправности  $T_{сун}$  с помощью следующей зависимости:

$$k_r(t) = \frac{T_{іі}}{T_{іі} + T_{сун}} \quad (7)$$

Вероятность безотказной работы БВ

$$P(t) = [1 - P_{сб}(T)] [1 - P_{отк}(T)]. \quad (8)$$

Подставив выражение (7) и (8) в (6), получаем

$$W = \frac{[1 - P_{сб}(T)][1 - P_{ітк}(T)]T_{іі}}{T_{іі} + T_{сун}} \quad (9)$$

Обобщающей характеристикой БВ является производительность  $E$  (или критерий эффективности применения вычислителя), под которой будем понимать количество решаемых задач от момента ввода программы и до выдачи команды управления в единицу времени.

В соответствии с методикой [8] для оценки эффективности определим время решения тестовой (эталонной) задачи:

$$T_3 = T_{вв}(1 - \varepsilon_1) + T_c + T_z + T_k + T_p + T_{выв}(1 - \varepsilon_2), \quad (10)$$

где  $T_{вв}$  – время ввода программы;

$T_c$  – время работы БВ при управлении;

$T_r$  – время, расходуемое на обмен информации между запоминающими устройствами;

$T_{выв}$  – время выдачи команд управления;

$T_k$  – время контроля функционирования БВ (в процессе решения задачи);

$\varepsilon_1$  и  $\varepsilon_2$  – коэффициенты совмещения ввода и вывода информации с процессом решения;

$T_p$  – время профилактического осмотра и ремонта.

Если предположить, что  $T_p = \beta T_3$ , то критерий можно записать как

$$E_p = \frac{1 - \beta}{T_{вв}(1 - \varepsilon_1) + T_c + T_z + T_k + T_{выв}(1 - \varepsilon_2)}. \quad (11)$$

Зная количество исходных чисел  $N_1$ , чисел в программе  $N_{пр}$ , чисел в выходной информации  $N_2$ , а так же  $V$ ,  $V_{вв}$  и  $V_{выв}$ , найдем следующие временные параметры:

$$\left. \begin{aligned} T_{\text{вб}} &= \frac{N_1 + N_{np}}{V_{\text{вб}}}; \\ T_c &= \frac{N_c}{V}; \\ T_{\text{вбвб}} &= \frac{N_2}{V_{\text{вбвб}}}; \end{aligned} \right\}, \quad (12)$$

где  $N_c$  – количество операций, выполняемых БВ в тестовой программе.

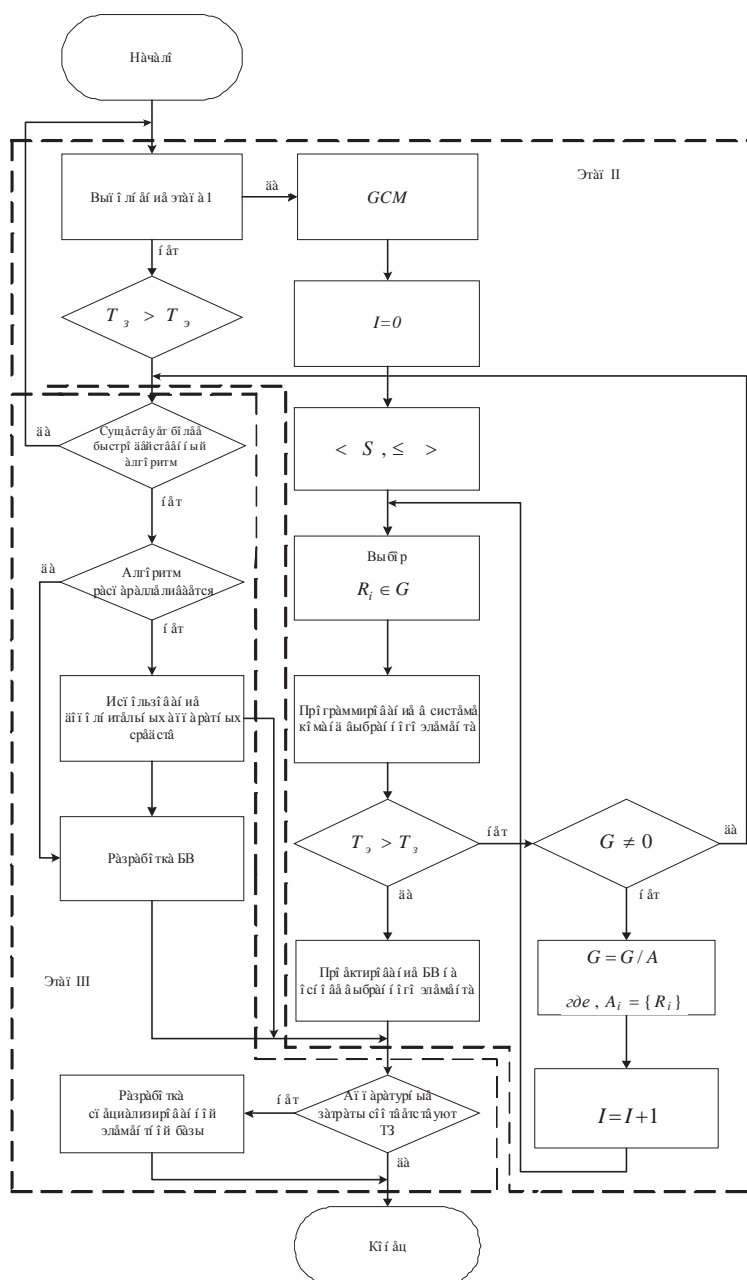


Рис.1. Алгоритм процедуры выбора элементной базы для БВ

Следовательно, время обмена информации можно записать в виде:

$$T_z = \sum_{(v)} [(Z_{1v} + Z_{2v})\tau_v + (S_v + S_v^*) \frac{\tau_{ov}}{2}] \frac{1}{D_v}, \quad (13)$$

где  $Z_{1v}$  – количество чисел и команд, пересылаемых в оперативное запоминающее устройство (ЗУ) из  $v$ -го постоянного ЗУ;

$Z_{2v}$  – количество чисел и команд, пересылаемых из ОЗУ в  $v$ -е ПЗУ;

$S_v, S_v^*$  – соответственно номера обращения к  $v$ -м ПЗУ для считывания или записи информации в ОЗУ;

$\tau_v, \tau_{ov}$  – время (чисел) работы соответственно ОЗУ и ПЗУ;

$D_v$  – коэффициент выигрыша времени при совмещении передачи информации из ОЗУ в ПЗУ и наоборот.

Подставив в выражение (11) формулы (12) и (13), определим

$$E = \frac{1 - \beta}{\frac{N_1 + N_{np}}{V_{66}(1 - \varepsilon_1)} + \frac{N_c}{V} + \sum_{(v)} [(Z_{1v} + Z_{2v})\tau_v + (S_v + S_v^*) \frac{\tau_{ov}}{2}] \frac{1}{D_v} + T_k + \frac{N_2}{V_{616}}(1 - \varepsilon_2)}. \quad (14)$$

Необходимо отметить, что критерий эффективности использования БВ определяется для нескольких режимов управления полетом.

Теперь переходим к выбору элементной базы для БВ. Предложенную методику выбора и проектирования БВ условно можно разбить на три этапа.

Этап I – предварительная обработка выбранного алгоритма на ЭВМ и получение оценочных временных характеристик выполнения программы на соответствующей элементной базе и ее оценки по выражениям (1):(14).

Этап II – выбор элементной базы для БВ, удовлетворяющей требованиям ТЗ. По предварительным оценкам (этап I) выбирает элементную базу с минимальными аппаратными затратами и уточняют время реализации алгоритма на выбранной элементной базе.

Этап III – проектирование БВ. Анализируется возможность использования более быстродействующих алгоритмов. Если такой существует, переходим к этапу I, иначе исходя из ТЗ и выбранного алгоритма, предлагается либо проектировать БВ, либо заказывать специализированную элементную базу, или за счет введения дополнительного оборудования часть функций алгоритма выполняемых программно, реализовывать аппаратно.

Рассмотрим более подробно каждый из этапов.

Этап I состоит из семи блоков, выполняемых последовательно (рис. 2).

После написания исходной программы на языке высокого уровня нумеруются операторы исходного модуля и в него включается отлаживающая программа “Шлях”, которая в процессе выполнения выдает на печать последовательность меток, имеющих в анализируемой программе.

Поэтому предварительно нумеруются все операторы исходного модуля.

Анализируемая программа транслируется в объектный модуль и реализуется на ЭВМ. По завершении выполнения программы полученная информация отражает число обращений к тому или иному оператору исходного модуля.

Затем анализируется объектный модуль, т.е. определяется соответствие оператора исходного модуля, написанного на языке высокого уровня, оператору

или ряду операторов объектного модуля, реализованным на языке АССЕМБЛЕР. Так как частота обращений к соответствующим операторам исходного модуля в процессе его выполнения известна из программы “Шлях”, то легко рассчитать число соответствующих коротких операций, имеющихся в программе

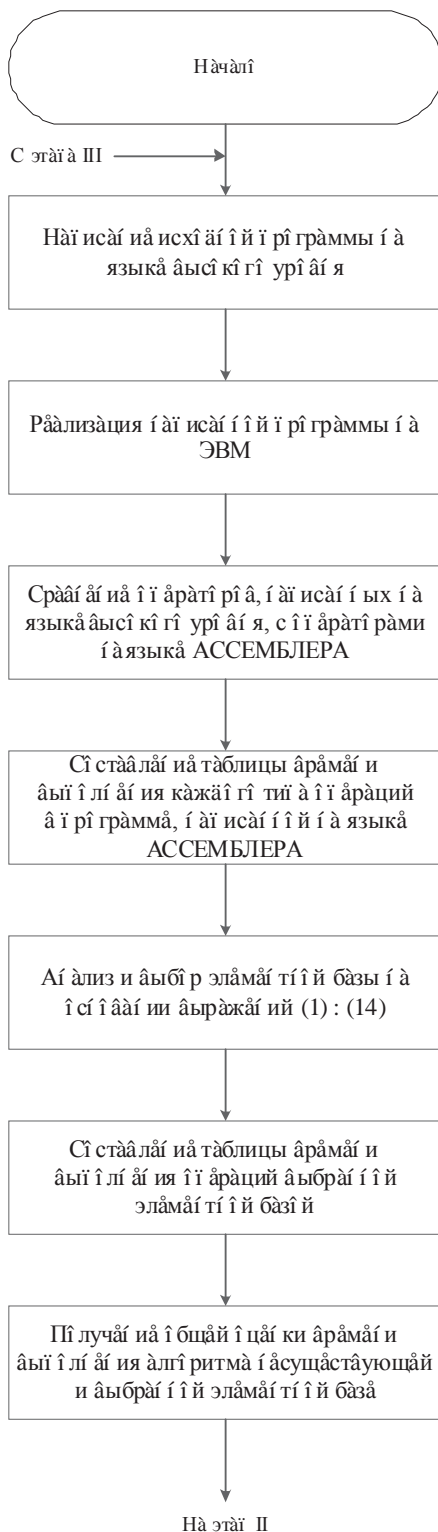


Рис. 2. Алгоритм процедуры выполнения этапа I

Все операторы АССЕМБЛЕРА объединены по типу операций в девять групп и представлены в виде таблицы. Аналогично составляется таблица, содержащая времена выполнения каждого типа операций на различных типах элементной базы, которая выбрана и проанализирована на основании выражений (1):(14). Перемножив соответствующие графы таблицы, а затем просуммировав результаты по каждому набору элементов базы, получим общую оценку времени выполнения алгоритма на существующей элементной базе.

Таким образом, на этапе I методики получены статистические данные о времени выполнения алгоритма при различных исходных данных и найдена максимальная оценка времени, исходя из быстродействия и реализуемого алгоритма.

Временные оценки, полученные на этапе I, не совсем точны, поскольку исходные программы пишутся не в системе команд, используемой элементной базы, а на языке высокого уровня, и объектный код, который анализируется на данном этапе, получен через транслятор, вносящий основную погрешность и избыточность. Поэтому для различных алгоритмов оценки, рассчитанные на этапе I, носят сравнительный характер и бывают в два ряда больше или меньше истинных, рассчитанных при программировании в системе команд элементной базы.

Для определения точных оценок по временным и аппаратным затратам переходим к этапу II, который реализуется следующим образом:

1. По рассчитанным на этапе I оценкам и показателям из всего множества элементной базы  $M$  составляется подмножества  $G \in M$ , удовлетворяющее по быстродействию требования ТЗ  $T_3$ .

2. Подмножества  $G$  частично упорядивается по критерию минимальных аппаратных затрат  $\langle G, \leq \rangle$  [9] и из него выбирается первый элемент  $R_1 \in G$ .

3. Для найденной элементной базы в системе его команд разработчик составляет программу и оценивает ее по быстродействию. Если полученная оценка удовлетворяет требованиям ТЗ, то переходим к проектированию БВ, реализованного на основе выбранной элементной базы, и к уточненной проверке аппаратных затрат (этап III, п. 2), иначе из подмножества  $G$  удаляется элемент

$$G \frac{\infty}{1} / R_i .$$

4. Если  $G \neq \emptyset$ , из него выбирается следующий элемент  $R_{i+1}$  и процедура повторяется, начиная с п. 3, иначе осуществляется переход к этапу III.

Этап III методики практически мало отличается от известных методик [3; 4; 5; 6; 9]. Он выполняется с самого начала лишь в том случае, если ни один из



предложенных наборов элементной базы не удовлетворяет по быстродействию ТЗ. Следовательно, необходимо проверить и проанализировать возможность использования других более быстродействующих алгоритмов. Если такие алгоритмы существуют, необходимо повторить все расчеты сначала, т.е. перейти к этапу I, иначе либо разрабатывается БВ при возможности распараллеливания алгоритма, либо используются дополнительные аппаратные затраты, позволяющие некоторые функции, выполняемые программно, реализовать аппаратно.

Затем сравнивают аппаратные затраты, полученные при проектировании с указанными в ТЗ. Если аппаратные затраты не удовлетворяют ТЗ, то рекомендуется разрабатывать заказную специализированную элементную базу, обеспечивающую заданное быстродействие и аппаратные затраты. Если аппаратные затраты не превышают объема и весо-габаритных характеристик аппаратуры, заданных в ТЗ на БВ, процесс выбора элементной базы и проектирования БВ заканчивается.

### Выводы

Предположенная методика выбора элементной базы для БВ беспилотного летательного аппарата и проектирование его позволяет анализировать программу на языке высокого уровня и носит универсальный характер, что упрощает процедуру выбора элементной базы, так как нет необходимости учитывать системы команд всех наборов элементов базы, и как в следствие уменьшается время проектирования БВ. Методика предусматривает возможность анализа всего алгоритма, а не выбранной его части, как в методе банчмарковских программ, обеспечивает высокую достоверность полученных результатов. В процессе работы и отладки алгоритма на ЭВМ параллельно с получением искомым оценок проверяется работоспособность предположенных алгоритмов.

### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Алексеев С.В.* Беспілотні літальні засоби: історія та перспективи розвитку / С.В. Алексеев // Сучасна спеціальна техніка. – № 3 (38). – 2014. – С. 89–98.
2. *Хорошко В.А.* Система обработки видеoinформации поступающей с беспилотников / В.А. Хорошко, Ю.Е. Хохлачева // Сб. науч. трудов НАУ “Защита информации”. – Вып. 22. – 2015. – С. 60–74.
3. *Кофронн Дж.* Технические средства микропроцессорных систем. Практический курс / Дж. Кофронн. – М. : Мир, 1983. – 190 с.
4. *Трояновский В.М.* Применение микропроцессоров и микроЭВМ / В.М. Трояновский. – М. : Высшая школа, Микропроцессоры. – Вып. 5. – 1998. – 160 с.
5. *Соботка З.* Микропроцессорные системы. Изд. 2-е / З. Сobotка, Я. Стары. – М. : Энергоиздат, 1992. – 496 с.
6. Специализированные ЦВМ: Учебник для вузов / В.Б. Смолов, В.В. Барященков, В.Д. Байков и др. Под ред. В.Б. Смолова Изд. 3-е. – М. : Высшая школа, 2001. – 309 с.

7. *Креденцер Б.П.* Техническое обслуживание и надежность систем с временным резервированием / Б.П. Креденцер. – К. : Феникс, 2016. – 384 с.
8. Основи надійності інформаційних систем / С.М. Головань, О.В. Корнейко, О.С. Петров та ін. – Луганськ : Вид-во “Ноулідж”, 2012. – 335 с.
9. МикроЭВМ / Пер. с англ. Под ред. А. Дирксона. Изд. 2-е. – М. : Энергоиздат, 2001. – 328 с.

Отримано 02.12.2016

Рецензент Рибальський О.В., д.т.н