

# Конкурсы по разработке программного и аппаратного обеспечения

## в рамках Гомельской виртуальной недели компьютерных наук

Михаил Долинский

dolinsky@gsu.unibel.by

С 1997 года ежегодно во второй декаде марта на базе математического факультета Гомельского государственного университета проводится Неделя компьютерных наук (Gomel Computer Science Week — GCSW). Стратегической целью GCSW является создание «технопарка», включающего в себя школы, вузы, научно-исследовательские организации, производственные предприятия и коммерческие фирмы, работающие в сфере новых компьютерных технологий. Естественным образом представляется и распределение интересов и обязанностей в таком технопарке:

- вузы готовят специалистов для научно-исследовательских организаций, производственных предприятий и коммерческих фирм, работающих в сфере новых компьютерных технологий, ориентируясь на приоритеты, определяемые «потребителями» специалистов;
- школы готовят ребят к поступлению в вузы и успешной учебе в них, ориентируясь на требования вузов к подготовке абитуриентов.

В этой парадигме GCSW отводится роль камертона, по которому настраивались бы все предполагаемые партнеры технопарка.

### GCSW '97 — с места в карьер

Уже в 1997 году программа GCSW '97 включала в себя следующие мероприятия:

- I. Олимпиада по программированию, 2 индивидуальных тура и 1 командный тур для студентов 1–5 курсов, школьников 10–11 классов, школьников 7–9 классов.
- II. Gomel Soft '97, конкурс программных продуктов, разработанных в Гомеле.
- III. Презентации фирм, связанных с компьютерными технологиями.
- IV. Научно-техническая конференция «Компьютерные науки: вчера, сегодня, завтра» с конкурсом на лучший научный доклад.
- V. Дискуссионный клуб «Перспективы Internet в Гомельской области».
- VI. Научно-методический семинар «Компьютерные технологии в образовании».
- VII. Выставка-продажа научно-технической литературы, CD-дисков, программного обеспечения, компьютеров и комплектующих.
- VIII. День открытых дверей математического факультета.

И все они были успешно проведены, кроме выставки-продажи с участием фирм. Удалось органи-

зовать лишь презентации нескольких гомельских фирм, таких, как ПКФ «Сервер» и ГП «Гомельоблтелеком».

В 1998 году программа GCSW сохранилась, и, кроме того, в первый (и пока последний) раз удалось провести выставку с участием известных на гомельском и белорусском рынке фирм в области внедрения новых компьютерных технологий: «Таир», «Галакта», «Спринкс», «Асбис», «Компьютеры и периферия», «Шанц», «Дайнова», «Сапрмаш», «Элмис» и «Технотроник». Значительную роль в том, что эта выставка состоялась, сыграли члены оргкомитета GCSW '98 — студенты 5-го курса математического факультета Владимир Гурик и Андрей Сапунов и студентка 3-го курса математического факультета Ирина Реут.

Кроме того, в 1998 году научно-техническая конференция в рамках GCSW получила одобрение Министерства образования Республики Беларусь и, как следствие, статус Республиканской научно-технической конференции для студентов, аспирантов и молодых ученых, а также средства на издание материалов конференции.

Не стал исключением в поступательном развитии GCSW и 1999 год. При этом основной особенностью GCSW '99 стал полноправный республиканский уровень, поскольку в мероприятиях GCSW '99 в массовом порядке принимали участие школьники, студенты и аспиранты практически из всех областей Белоруссии.

Программа GCSW '99 включала в себя следующие мероприятия:

- I. Республиканская научно-техническая конференция для студентов и аспирантов.
- II. Личная олимпиада по программированию для школьников, начинающих изучение программирования.
- III. Командная олимпиада по программированию для школьников, начинающих изучение программирования.
- IV. Личная олимпиада по программированию для школьников, продолжающих изучение программирования.
- V. Открытое командное первенство вузов республики (только в очной форме олимпиады участвовали по 2 команды от БГУИР, ГрГУ, БрГУ, БелГУТ, 5 команд от ГГУ и 8 команд школьников).
- VI. Заочный и очный конкурсы по решению шахматных задач.
- VII. Заочный и очный туры олимпиады по занимательной математике.

**VIII. Конкурс по проектированию цифровых устройств**, в котором кроме команд ГГУ приняла участие команда Томского политехнического института (по электронной почте).

Необходимо отметить, что во всех олимпиадах и конкурсах важной составляющей было обеспечение удаленного участия в олимпиадах и конкурсах. А в студенческом командном турнире по программированию впервые в республике Беларусь было обеспечено удаленное тестирование в режиме реального времени, то есть 2 команды, которые во время турнира находились непосредственно в Минске и 1 — в Гродно, участвовали точно так же, как и те 20 команд, которые располагались в компьютерных классах университета.

Другой важной особенностью олимпиады была практически полная автоматизация проверки присланных решений для всех конкурсов и олимпиад, иначе мероприятие такого масштаба просто невозможно было бы организовать, ведь всего в индивидуальных и командных соревнованиях приняло участие около 250 школьников и 40 студентов из всех областей республики.

Недостатком GCSW '99 по сравнению с GCSW '98 необходимо признать отмену проведения выставки — нам просто не удалось ее организовать.

Важно отметить также, что Оргкомитет GCSW'99 в полной мере осознал необходимость полной автоматизации организационной работы с участниками для возможности дальнейшего увеличения их количества.

Уже с начала подготовки GCSW '2000 мы активно задействовали сайт GCSW (<http://www.gsu.unibel.by/gcsw2000>). Там были размещены как анонсы всех мероприятий, планируемых на GCSW '2000, так и архивы 3-х предыдущих форумов (GCSW '97, GCSW '98, GCSW '99).

Надо отметить, что интенсивное использование сайта началось с 1998 года, когда на нем были размещены анонсы мероприятий, карта университетского кампуса, сведения о фирмах-участниках выставки и краткие сведения о GCSW '97. В 1999 году на сайте дополнительно были размещены списки участников всех мероприятий со ссылками на их адреса и e-mail, задания и результаты олимпиад, материалы докладов на конференции.

Принципиальное отличие сайта GCSW '2000 — максимальная автоматизация работы с пользователями, начиная с регистрации. Одновременно надо отметить еще несколько фактов, повлиявших на принятие «революционного» решения, о котором речь пойдет дальше:

- уже в 1999 году было понятно, что энтузиазм активистов оргкомитета GCSW «на пределе». И дальнейшее распространение сферы действий GCSW (а оно предполагалось перманентным с самого начала) невозможно;
- с октября 1999 года на сайте ГГУ стартовал новый проект «Дистанционное обучение в Беларуси» ([dl.gsu.unibel.by](http://dl.gsu.unibel.by)).

Сам проект заслуживает отдельной статьи. Здесь же замечу, что он, среди прочих своих задач, поддержал также и полную автоматизацию проведения всех олимпиад и конкурсов,

проводимых в рамках GCSW с 2000 года, начиная от регистрации участников и заканчивая отсылкой им конечных результатов. При этом такой автоматизацией охвачены не только те пользователи, которые имеют доступ в Internet on-line, но и те, у которых есть только электронная почта, выход в сеть ФИДО или даже просто персональный компьютер с модемом!

И вот все эти факты привели к принятию Оргкомитетом GCSW стратегического решения: GCSW в перспективе должна стать «виртуальной», а следом и «перманентной». Что здесь имеется в виду?

- Основные события GCSW не должны требовать от участников обязательного приезда в Гомель.
- Хотя в марте возможно и останется какая-то фокусировка, акцентация, но собственно процесс взаимодействия и взаимовлияния участников GCSW должен длиться в течение всего года.

### GCSW 2000-200x — как это происходит

Как проходят мероприятия GCSW (ежегодно, во второй декаде марта на полной рабочей неделе), ориентируясь на новые парадигмы «бытия»:

- возможность участия во всех мероприятиях GCSW без физического перемещения в Гомель;
- оперативность, доступность и привлекательность информации обо всех мероприятиях GCSW не только для непосредственных участников данного мероприятия, но и для наблюдателей;
- полная автоматизация проведения всех мероприятий GCSW с целью снять ограничение на количество участников в любом мероприятии.

#### I. Личная олимпиада по программированию для профессионалов

По традиции основной контингент этих соревнований — школьники, готовящиеся к республиканской олимпиаде по информатике, которая проводится спустя несколько недель после нашей олимпиады. Тем не менее, эта олимпиада открыта для всех: школьников, не попавших на республиканскую олимпиаду текущего года по результатам областных олимпиад; студентов, скучающих по «личным» соревнованиям; профессионалов, не боящихся сразиться в честном бою со школьниками; и вообще всех желающих.

#### II. Личная олимпиада по программированию для начинающих

По задумке организаторов эта олимпиада предназначена для тех, кто изучает программирование в течение месяца-трех, и соответственно вовлекающая в соревнования учеников 6-8 классов. Однако практика показывает, что в них с удовольствием принимают участие и школьники более старшего возраста, по разным причинам не готовые участвовать в олимпиадах по программированию для профессионалов.

Надо отметить, что обе описанные олимпиады — личные и проводятся в 2 тура. То есть с 8 утра во вторник и в среду условия

задач 1-го и 2-го тура соответственно выкладывались на сайт олимпиады. Кроме того, они рассылались всем, кто зарегистрировался на сайте и указал свой электронный почтовый адрес. 5 часов участники решали задачи (соблюдение этого правила на совести участника), затем отсылали свое решение на сайт, где оно автоматически тестировалось, и результаты тестирования отсылались участнику, одновременно пополняя и сводную таблицу олимпиады на сайте. По завершении каждого дня результаты по запросу (посредством Internet или e-mail) могли быть получены любым участником олимпиады.

Следующие 2 олимпиады командные, обе проводятся одновременно в четверг, с 9.00 до 14.00 по поясному времени GMT+2.

#### III. Командная олимпиада по программированию для профессионалов

Эта олимпиада ориентирована на команды вузов, и проводится она по правилам командного студенческого первенства мира по программированию, которое ежегодно организуется ACM — международной ассоциацией компьютерных специалистов.

В кратком изложении правила таковы:

- условия на английском языке включают от 6 до 8 задач;
- в команде должно быть ровно 3 человека и один компьютер;
- проверка решений осуществляется в режиме реального времени: команда отправляет решение не в конце олимпиады, как это традиционно происходит в случае личных олимпиад по информатике для школьников, а как только участникам показались, что команда решила какую-то из предложенных задач. Сразу по приходу решения, в порядке «живой» очереди, тестирующая система принимается за проверку присланного решения, и по завершении тестирования отправляет команде результат проверки, пополняя протокол тестирования и обновляя текущую таблицу результатов, доступную всем участникам;
- и наконец, последняя важная особенность правил — в этих соревнованиях засчитываются только полные решения. То есть решения, которые выдают правильные результаты на всех тестах, подготовленных жюри.

Таким образом, побеждает команда, которая полностью решит большее количество задач. В случае равенства по количеству решенных задач сравнивается суммарное потраченное время на решение задач. Суммарное время решения задач командой равно сумме времени по каждой принятой задаче. А время по принятой задаче складывается из времени от начала соревнований до времени приема задачи, а также штрафного времени, начисляемого по формуле  $(20 \text{ мин}) \times (N-1)$ . Где N — номер попытки сдачи решения, которая оказалась успешной. Из формулы следует, что в случае успешной сдачи с первой попытки штрафное время по этой задаче не начисляется.

И хотя олимпиада ориентирована на подготовку команд белорусских вузов к командному студенческому первенству мира по программированию, в ней нет ограничений

на участие — для школьников, профессиональных программистов и других любителей таких соревнований.

И действительно, эта олимпиада является одним из наиболее эмоциональных мероприятий GCSW.

#### IV. Командная олимпиада по программированию для начинающих

Отличается от олимпиады для профессионалов только тем, что условия на русском языке и задания значительно проще.

#### V. Олимпиада по информатической математике

Изначально эта олимпиада была задумана как средство привлечь к Неделе компьютерных наук младших и средних школьников (4–8 классы), еще не умеющих программировать, с целью отобрать из участников этой олимпиады ребят, склонных и способных к изучению программирования.

Мучительная подготовка условий задач прошлогодней первой олимпиады такого рода привела вначале к выработке парадигм олимпиады:

- в задаче должно быть жизненное развернутое условие**, заставляющее мыслителей «отделить зерна от плевел», то есть выяснить, что в условиях задачи является ее художественным оформлением, а что определяет математическую суть задачи;
- решение предполагает исследование, разработку и исполнение алгоритма**. Многолетний опыт в работе с начинающими изучение программирование показал, что именно отсутствие этого навыка у обучаемого, а не сложность абстракций и конструкций языков программирования становится основным камнем преткновения для большинства пытающихся изучать программирование;
- однозначный автоматически проверяемый ответ** — с целью снятия ограничений на количество участников олимпиады за счет использования автоматической системы проверки решений;
- задачу могут решить самые маленькие** — опять-таки опыт работы показал, что алгоритмические навыки не обязательно определяются возрастом ребенка. И потому хотелось вовлечь как можно более широкий круг ребят.

Сопоставление всех этих требований привело к решению, которое на первый взгляд показалось оригинальным, а на все последующие взгляды — простым, естественным и единственно соответствующим как парадигмам составления задач, так и глобальной цели этой олимпиады.

На олимпиаде по информатической математике выдаются условия задач те же самые, что и на олимпиаде по программированию для начинающих!

Разница будет заключаться в том, что на олимпиаде по информатической математике участники вместе с условиями задач получают еще и входные данные тестов. Задача участников олимпиады — правильно вычислить (можно без написания программ) выходные результаты тестов. И именно эти результаты сообщить в жюри для каждого просчитанного теста каждой задачи.

Поскольку задачи рассчитаны на самый широкий круг участников, сделаны и некоторые послабления:

- в соревнованиях могут участвовать как отдельные ученики, так и команды из учеников;
- не регламентируется количество участников команды, оно должно определяться участниками или их учителем из соображений максимальной занятости каждого участника в течение олимпиады;
- допускается помощь учителей, родителей и руководителей команд в организации работы команд и разъяснении фактов и сведений, необходимых для решения задач, не известных участникам команды;
- оценка проводится по баллам, начисляемым за каждый верно вычисленный ответ.

#### VI. Конкурс по решению шахматных задач

На трое суток — с 9.00 утра вторника до 9.00 утра пятницы открываются условия нескольких десятков шахматных задач от самых простых (мат в один ход), до достаточно сложных многоходовых этюдов. Условие каждой задачи представлено в виде рисунка шахматной доски и перечня позиций. Решение (полный набор всех вариантов) можно вводить либо текстовой записью ходов, либо интерактивным перемещением фигур на доске.

#### VII. Конкурс по английскому языку

На те же трое суток со вторника по пятницу открывается набор упражнений по английскому языку.

#### VIII. Конкурсы по проектированию цифровых устройств, разработке программ для микроконтроллеров, совместной разработке программного и аппаратного обеспечения встроенных систем

Появление этих конкурсов связано с одной из ведущих прикладных тематик, разрабатываемых на математическом факультете Гомельского государственного университета — «Автоматизация разработки программного и аппаратного обеспечения встроенных систем» ([newit.gsu.unibel.by](http://newit.gsu.unibel.by)). Для этих конкурсов, так же как и для конкурса по решению шахматных задач, условия объявляются во вторник утром, а тестирование решений прекращается в пятницу утром — ровно через 3 суток после обнародования условий задач.

Поскольку основной целью данной статьи как раз и является привлечение внимания читателей журнала именно к этим конкурсам, далее о них будет рассказано более подробно.

### Конкурс по проектированию аппаратного обеспечения цифровых устройств

#### Правила

Во время соревнований команды решают предложенные задачи. Команда может решать задачи только с помощью системы проектирования HLCCAD. Решением является проектный файл или набор файлов в ZIP-архиве. Устройству дается не более 100 ps для реакции на воздействие, после этого подается следующее, либо проверяются значения на выходах.

Команда имеет право отправить проект в жюри на тестирование сразу, как только ко-

манде кажется, что проект завершен. О результатах тестирования команде сообщается в течение нескольких минут. Каждая неудачная попытка тестирования увеличивает время выполнения проектов на 20 минут.

Побеждает команда, правильно выполнившая большее количество проектов. В случае одинакового количества выполненных проектов сравнивается суммарное время, ушедшее на разработку проектов. Для каждого проекта время его выполнения отсчитывается от начала соревнований.

Проверка представленных решений проводится во время соревнований. В присланном наборе файлов должен находиться файл с именем, оговоренным в задании и содержащим в корне проекта устройство, имя и формат которого также указаны в задании.

Решение проверяется путем запуска на наборе тестов, который недоступен участникам и является одинаковым для всех команд. Решение засчитывается только в том случае, если оно выдает верные ответы на все тесты. Все входные данные предполагаются корректными и удовлетворяющими всем ограничениям, указанным в условии.

#### Примеры заданий

##### 1) Задание «Cuba»

Имя HLCCAD-проекта: GCSW2002.PRD

Входное устройство: Cuba

| Название | Размерность | Тип   |
|----------|-------------|-------|
| In       | 20          | Вход  |
| Out      | 7           | Выход |

Целое положительное число (от 0 до 99) возвели в куб и это значение подали на вход устройства. На выходе устройства необходимо определить искомое число.

#### Пример:

In = 10101100000111001001

Out = 1011001

##### 2) Задание «BarCodeChecker»

Имя HLCCAD-проекта: GCSW2001.PRD

Входное устройство: BarCodeChecker

| Название | Размерность | Тип   |
|----------|-------------|-------|
| Code1    | 4           | вход  |
| Code2    | 4           | вход  |
| Code3    | 4           | вход  |
| Code4    | 4           | вход  |
| Code5    | 4           | вход  |
| Code6    | 4           | вход  |
| Code7    | 4           | вход  |
| Code8    | 4           | вход  |
| Code9    | 4           | вход  |
| Code10   | 4           | вход  |
| Code11   | 4           | вход  |
| Code12   | 4           | вход  |
| Code13   | 4           | вход  |
| Status   | 1           | выход |
| Error    | 1           | выход |

Устройство для проверки 13-разрядного штрих-кода системы EAN-13.

Пусть дан штрих-код «4601546003119».

Для проверки правильности штрих-кода существует следующий алгоритм:

1. Складываются цифры, стоящие на четных местах штрих-кода  
 $6+1+4+0+3+1=15$
2. Полученная сумма умножается на три  
 $15 \times 3 = 45$

3. Складываются цифры, стоящие на нечетных местах штрих-кода (кроме последней, контрольной цифры)  
 $4+0+5+6+0+1=16$

4. Полученные два числа складываются  
 $45+16=61$

5. Отбрасываются десятки  
 $61-60=1$

6. Полученное в пункте 5 число вычитается из десяти  
 $10-1=9$  (если в 5 пункте получили 0, результат — 0)

Результат должен совпадать с контрольной цифрой (13-ой цифрой). Если это не так, то указанный штрих-код — грубая подделка, а качество товара, скорее всего, невысоко.

Code1 — Code12 — первые 12 цифр штрих-кода.

Code13 — 13-я цифра штрих-кода (контрольная).

Error — некорректные входные данные (1 — данные некорректны).

Status — результат проверки (1 — штрих-код неверен или некорректные входные данные)

#### Пример:

«4601546003119» — штрих-код верен.

«4601546003118» — штрих-код неверен.

#### 3) Задание «Stack»

Имя HLCCAD-проекта: GCSW2001.PRD  
 Входное устройство: Stack

| Название | Размерность | Тип   |
|----------|-------------|-------|
| Input    | 8           | вход  |
| Push     | 1           | вход  |
| Pop      | 1           | вход  |
| Reset    | 1           | вход  |
| Output   | 8           | выход |
| Count    | 8           | выход |
| Error    | 1           | выход |

Модель 8-разрядного стека с глубиной до 255 элементов.

Позволяет:

- положить элемент в стек;
- взять значение вершины стека;
- убрать элемент из стека;
- узнать количество элементов в стеке;
- контролировать ошибки (попытка взять элемент из пустого стека);
- сбросить состояние стека и признак ошибки.

Input — значение элемента.

Output — значение вершины стека.

Count — количество элементов (для пустого стека — 0).

Reset — сброс стека (сброс при 1).

Push — положить элемент с Input в стек (по переднему фронту).

Pop — убрать элемент со стека (по переднему фронту).

Error — признак ошибки (1 — ошибка).

### Конкурс по разработке программ для микроконтроллеров

#### Правила

В данном конкурсе предлагается написать программы на ассемблере микроконтроллеров AVR, i8051 и M68HC08. Если все эти микрокон-

троллеры не знакомы, то можно решать задачи на ассемблере для i8086. Что бы уравнивать шансы участников, программирующих решения на ассемблере для микроконтроллеров, и участников, программирующих решения на ассемблере i8086, в модели этого процессора оставлен только один сегмент размером 64 кбайт, отсутствует поддержка цепочных инструкций и инструкций с плавающей точкой.

#### Механизм передачи исходных данных и получения результатов

Перед выполнением тестируемого решения переменным, указанным в условии задачи как переменные с исходными данными, присваиваются определенные значения. Затем происходит выполнение программы до последней строки. После этого значения переменных, указанных в условии как результирующие, сравниваются с эталоном и определяется, прошла задача тест или не прошла. Последняя строка программы определяется по наличию в строке с инструкцией комментария ;\$E.

#### Пример оформления задачи

Условие задачи: найти сумму двух чисел. Первое число задается переменной N1, второе число задается переменной N2, результат должен быть в переменной R.  $0 \leq N1, N2, R \leq 255$ .

Например: N1 = 23, N2 = 1, R = 24.

Тогда решение задачи i8051 оформляется так:

```

dseg
org 60h
N1: ds 1
N2: ds 1
R: ds 1
cseg
org 0h
sjmp start
org 30h
start:
mov a, N1
mov b, N2
add a, b
mov R, a
halt:
sjmp halt,$E

```

Данное решение должно быть отослано на тестирование в файле с расширением i51.

Расширение у файла с решением в зависимости от типа микроконтроллера должно соответствовать следующей таблице:

| Тип платформы | Расширение файла |
|---------------|------------------|
| AVR           | AVR              |
| i8051         | i51              |
| i8086         | i86              |
| M68HC08       | m08              |

#### Пояснение сообщений о результатах тестирования

| Сообщение  | Расшифровка   |
|--|---|
| Пустая строка  | Задача прошла все тесты и в поле «Результат» указано количество заработанных баллов.  |
| Тест не прошел   | Задача не прошла один или несколько тестов.   |
| Тест не прошел по времени                              | Истекло отпущенное на исполнение время (миллион циклов микроконтроллера), а программа не дошла до последней строки.   |
| Ошибка в исходном тексте                               | Решение задачи содержит синтаксические ошибки или подготовлено для трансляции ассемблером, отличным от принятых на конкурсе (WInteg для соответствующей платформы). |
| Не указана в исходном тексте точка останова исполнения | В тексте решения отсутствует строка, помеченная как последняя комментарием ;\$E.  |

#### Примеры заданий

##### 1) Задание «число Фибоначчи»

Последовательность чисел Фибоначчи строится следующим образом: первые два числа равны 1, а каждое следующее равно сумме двух предыдущих. Задача — найти N-ое число Фибоначчи.

#### Формат ввода

N — номер числа Фибоначчи.

#### Формат вывода

K — значение N-го числа Фибоначчи.

Размер исходных данных и результатов — 1 байт.

| Пример ввода | Пример вывода |
|--------------|---------------|
| 3            | 2             |

##### 2) Задание «Программа А»

На входе дана строка, заканчивающаяся символом с кодом 0. Получить новую строку (также заканчивающуюся символом с кодом 0) путем удаления из исходной строки букв «А», «а».

#### Формат ввода:

st — исходная строка.

#### Формат вывода:

st1 — полученная строка.

#### Ограничения:

Исходная строка меньше 12 символов.

| Пример ввода | Пример вывода |
|--------------|---------------|
| NatAsha      | Ntsh          |

##### 3) Задание «Отнять»

Уменьшить каждую цифру числа на наименьшую цифру в его записи.

#### Формат ввода:

N1 — исходное число.

#### Формат вывода:

R — полученное число.

#### Ограничения:

$N1 < 255$ .

| Пример ввода | Пример вывода |
|--------------|---------------|
| 52           | 30            |

### Конкурс по совместной разработке программного и аппаратного обеспечения встроенных систем

#### Правила

Во время соревнований команды решают предложенные задачи. Команда может решать задачи только с помощью системы проектирования IEEED-2000. Решением является пара файлов: первый файл — исходный текст на ассемблере для микропроцессора, второй файл — проект в формате IEEED-2000 со схемой устройства (названия проекта и имя устройства указаны в условии задачи).

Команда имеет право отправить проект в жюри на тестирование сразу, как только команде покажется, что проект завершен. О результатах тестирования команде сообщается в течение нескольких минут. Каждая неудачная попытка тестирования увеличивает время выполнения проектов на 20 минут.

Побеждает команда, правильно выполнившая наибольшее количество проектов. В случае совпадения количества выполненных проектов сравнивается суммарное время, ушедшее на разработку проектов. Для каждого

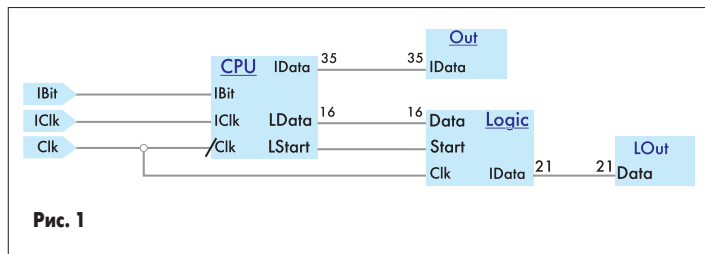


Рис. 1

проекта время его выполнения отсчитывается от начала соревнований.

Проверка представленных решений проводится во время соревнований. Решение проверяется путем запуска на наборе тестов, который недоступен участникам и является одинаковым для всех команд. Решение засчитывается только в том случае, если оно выдает верные ответы на все тесты. Все входные данные предполагаются корректными.

#### Примеры заданий

##### 1) Задание «Извлечение квадратного корня»

На вход IBit устройства последовательно подаются биты входного целого беззнакового 16-битного числа, начиная с младшего бита. Передача строится обратным фронтом сигнала, поступающего на вход ICk. Частота изменения стролирующего сигнала равна 50 кГц (рис. 1).

Проект в формате IEESD-2000 прилагается.

Блок Sqrt\_CPU преобразует число для отображения в десятичной системе счисления на группе семисегментных индикаторов блока Out. На получение и обработку числа устройству Sqrt\_CPU дается 10 мс. Кроме того, в его функции входит передача полученного числа на устройство Sqrt\_Logic. При этом все биты передаются одновременно по 16-разрядной шине LData.

Устройство Sqrt\_Logic преобразует число, полученное с входа Data по обратному фронту сигнала Start. На выходе IData необходимо получить число, равное целой части квадратного корня из Data для отображения в десяти-

чной форме на группе семисегментных индикаторов LOut.

На обработку числа устройству Sqrt\_Logic дается 10 мкс. После обработки одного числа устройства должны быть готовы к приему следующего.

На вход Clk поступают импульсы синхронизации устройств Sqrt\_CPU и Sqrt\_Logic с частотой 12 МГц.

Для Sqrt\_CPU, которое реализовано на базе микроконтроллера Intel 8051, требуется написать программу «Input.a51».

Для Sqrt\_Logic необходимо разработать схему.

**Решения принимаются** в ZIP-архиве, содержащем файл Input.a51 и файл Sqrt\_logic.PRD с устройством Sqrt\_logic.

##### 2) Задание «Количество единиц в сумме»

Устройство ADD\_CPU вычисляет сумму двух беззнаковых 16-битных чисел и передает результат на устройство ADD\_Logic, которое определяет количество единиц в нем (рис. 2).

Проект в формате IEESD-2000 прилагается.

На вход ID устройства ADD\_CPU последовательно подаются тетрады бит целых беззнаковых 16-битных чисел, начиная с младшей тетрады. Передача строится обратным фронтом сигнала, поступающего на вход DCk. Частота изменения сигнала равна 50 кГц. Устройство, реализованное на базе AT90S2313, находит сумму двух чисел и передает результат на блок ADD\_Logic потетрадно, начиная с младшей тетрады. Прием данных осуществляется по обратному фронту на выходе ODClk. На получение

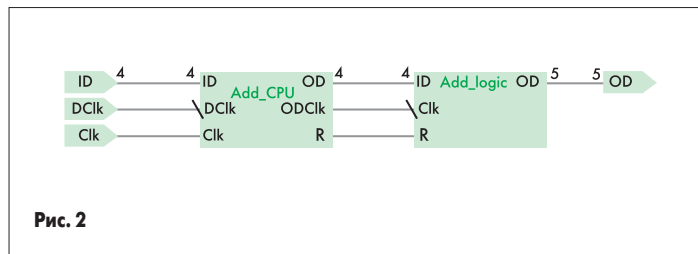


Рис. 2

и обработку числа устройству ADD\_CPU дается 100 мкс.

Устройство ADD\_Logic определяет количество единиц в полученном числе и выдает результат на выход OD. На обработку числа устройству ADD\_Logic дается 100 нс.

На вход Clk поступают тактовые импульсы с частотой 10 МГц.

Для ADD\_CPU, которое реализовано на базе микроконтроллера Atmel AT90S2313, требуется написать программу «Add.avg». Для ADD\_Logic необходимо разработать схему.

**Решения принимаются** в ZIP-архиве, содержащем файл Add.avg и файл Add\_logic.PRD с устройством Add\_logic.

#### Заключение

В рамках ежегодной Гомельской недели компьютерных наук наряду с подробно описанными конкурсами проводится также и Республиканская научно-техническая конференция студентов, аспирантов и молодых ученых «Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях».

Цель данной статьи — привлечь к GCSW внимание как участников, так и потенциальных спонсоров, в особенности для конкурсов по автономному и совместному проектированию аппаратного и программного обеспечения, которые, по нашему мнению, должны быть особенно интересны читателям журнала «Компоненты и технологии».