

Задания для проектов по компьютерной арифметики

1. Проект №1 "Логические операции над битовыми данными"

1) Разработать программную модель операции вида:

$$\begin{aligned} & \{[y((m-1), j_{(m-1)})]_{k_{y((m-1), j_{(m-1)})}}\}, j_{(m-1)} \in [0, l_{m-1} - 1] = \\ & = f_{((m-1)j_{(m-1)})} \{[y((m-2), j_{(m-2)})]_{k_{y((m-2), j_{(m-2)})}}\}, j_{(m-2)} \in [0, l_{m-2} - 1] \end{aligned} \quad (1)$$

Для $\forall \alpha, [y]_{k_y}$ справедливо:

$$\begin{aligned} & \{[y((m-1), j_{(m-1)})]\} = f_{((m-1)j_{(m-1)})} (\{[y((m-2), j_{(m-2)})]\}), j_{(m-2)} \in [0, l_{m-2} - 1]; \\ & \{[y((m-2), j_{(m-2)})]\} = f_{((m-2)j_{(m-2)})} (\{[y((m-3), j_{(m-3)})]\}), j_{(m-3)} \in [0, l_{m-3} - 1]; \\ & \dots \\ & \{[y(2, j_2)]\} = f_{(2j_2)} (\{[y(1, j_1)]\}), j_1 \in [0, l_1 - 1]; \\ & \{[y(1, j_1)]\} = f_{(1j_1)} (\{[y(0, j_0)]\}), j_0 \in [0, l_0 - 1] \end{aligned} \quad (2)$$

где $[y(i, j_i)]_{k_{y(i, j_i)}} \in \mathbb{Z}$ - число уровня i в системе счисления с основанием $k_{y(i, j_i)}$; j_i - номер числа в уровне i ; $j_i \in [0, 1, \dots, (l_i - 1)]$; $\{[y((m-1), j_{(m-1)})]_{k_{y((m-1), j_{(m-1)})}}\}, j_{(m-1)} \in [0, l_{m-1} - 1]$ - множество выходных чисел, их уровень равен $m-1$; $\{[y(0, j_0)]_{k_{y(0, j_0)}}\}, j_0 \in [0, l_0 - 1]$ - множество входных чисел, их уровень равен 0. Количество чисел на уровне № β - l_β .

2) Каждое число определяется как функция вида $f_{(i, j_i)}(y((i-1), j_{(i-1)})), f_{(i, j_i)} \in P_2$ - см. (2).

3) Перечень параметров модели:

Входные переменные (числа) $\{[y(0, j_0)]_{k_{y(0, j_0)}}\}, j_0 \in [0, l_0 - 1]$. Каждое число вида $y(0, j_0)$

является результатом работы генератора случайных чисел (ГСЧ), закон распределения — квази-равномерный. Разрядность каждого числа - r_y - также задается автоматически с помощью ГСЧ в ходе работы модели.

а) **Промежуточные числа** $[y(i, j_i)]_{k_{y(i, j_i)}}$. Для любого уровня параметр $i = \text{const}$, параметр

j_i является результатом работы ГСЧ, $j_i = 0, 1, \dots, l_i - 1$.

б) **Функции** $f_{(i, j_i)}(y((i-1), j_{(i-1)})), f_{(i, j_i)} \in P_2$. Количество функций в уровне № i равно

$l_i - 1$, параметр l_i является результатом работы ГСЧ. Конкретный вид функции (&

\oplus, \vee и т. д.) каждый раз задается как результат случайной выборки из базы данных функций двух переменных. Исключение составляют константы "0" и "1".

- с) **Основание числовой системы** для каждого из чисел. Определяется как результат работы ГСЧ, $k_{y(i, l_i)} \in \mathbb{Z}, 1 < k_{y(i, l_i)} \leq 10$.

Таким образом, выходные числа модели операции конструируются с помощью функций вида $[y(i, j_i)]_{k_{y(i, j_i)}}$, граф алгоритма построения выходных чисел является ориентированным,

ациклическим и представлен в виде ярусно-параллельной формы.

Пример работы модели операции над битовыми данными.

Исходные данные (ГСЧ). Система генерирует $m = 5$.

- **Уровень 0.** $i := 0$. Система: $l_0 = 10 \Rightarrow j_0 = 0, 1, \dots, 9$.

Система генерирует с помощью ГСЧ значения (k) оснований числовых систем и массив входных чисел:

$j_{(0)}$	0	1	2	3	4	5	6	7	8	9	/Исходные числа/
$\{k_{y_{(0, j_0)}}\}$	7	3	2	3	5	4	2	6	6	2	
$\{y_{(0, j_0)}\}$	3	1	0	2	3	2	1	4	5	1	

Назначение функций. В системе должна быть реализована база данных функций, номер каждой функции назначается с помощью ГСЧ. Пример:

№	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Функция	\vee	\wedge	\oplus	\equiv	\rightarrow	\downarrow	$ $	$\neg \rightarrow$	$\neg x$	x	y	\leftarrow	$\neg \leftarrow$	$\neg y$

- **Уровень 1.** $i := 1$. Генерация: $l_1 = 4, \Rightarrow j_1 = 0, \dots, 3$. Затем система генерирует функции

$f_{(1, j_1)}$, связанные с ними числа y_{1, j_1} и основания $k_{y_{1, j_1}}$:

$\{y_{1, j_1}\}$	$y_{(1, 0)}$	$y_{(1, 1)}$	$y_{(1, 2)}$	$y_{(1, 3)}$
$k_{y_{1, j_1}}$	4	3	3	2

Генерация функций:

$\{y_{0, j_0}\}$	$[3]_7$	$[1]_3$	$[0]_2$	$[2]_3$	$[3]_5$	$[2]_4$	$[1]_2$	$[4]_6$	$[5]_6$	$[1]_2$
------------------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

$[y_{(1,0)}]_4$	\vee	\vee						\vee		
$[y_{(1,1)}]_3$		\oplus		\oplus	\oplus	\oplus				
$[y_{(1,2)}]_3$			\wedge				\wedge	\wedge	\wedge	
$[y_{(1,3)}]_2$							\downarrow		\downarrow	\downarrow

Работа системы:

$$\begin{aligned}
 [y_{(1,0)}]_4 &= [3]_7 \vee [1]_3 \vee [4]_6 = [011]_2 \vee [001]_2 \vee [100]_2 = [111]_2 = [013]_4 \\
 [y_{(1,1)}]_3 &= [1]_3 \oplus [2]_3 \oplus [3]_5 \oplus [2]_4 = [001]_2 \oplus [010]_2 \oplus [011]_2 \oplus [010]_2 = [010]_2 = [002]_3 \\
 [y_{(1,2)}]_3 &= [0]_2 \wedge [1]_2 \wedge [4]_6 \wedge [5]_6 = [000]_2 \wedge [001]_2 \wedge [100]_2 \wedge [100]_2 = [000]_2 = [000]_3 \\
 [y_{(1,3)}]_2 &= [1]_2 \downarrow [5]_6 \downarrow [1]_2 = [001]_2 \downarrow [101]_2 \downarrow [001]_2 = [010]_2
 \end{aligned}$$

Выходные значения $\{y_{1,j_1}\}$ должны быть сформированы в виде двух массивов:

$[\{y_{1,j_1}\}]_{k_{y_1,j_1}}$ и $[\{y_{1,j_1}\}]_2$. Это необходимо для проверки и контроля.

- **Уровень 2.** $i:=2$. Генерация: $l_2 = 6, \Rightarrow j_2 = 0, \dots, 5$. Затем система генерирует функции

$f_{(2,j_2)}$, связанные с ними числа y_{2,j_2} и основания $k_{y_{2,j_2}}$:

$\{y_{2,j_2}\}$	$y_{(2,0)}$	$y_{(2,1)}$	$y_{(2,2)}$	$y_{(2,3)}$	$y_{(2,4)}$	$y_{(2,5)}$
$k_{y_{2,j_2}}$	7	3	4	5	2	3

Генерация функций:

$\{y_{1,j_1}\}$	$[013]_4$	$[002]_3$	$[000]_3$	$[010]_2$
$[y_{(2,0)}]_7$	\equiv		\equiv	
$[y_{(2,1)}]_3$		\wedge		\wedge
$[y_{(2,2)}]_4$	\rightarrow			\rightarrow
$[y_{(2,3)}]_5$	$ $	$ $	$ $	
$[y_{(2,4)}]_2$	$\neg \rightarrow$		$\neg \rightarrow$	$\neg \rightarrow$
$[y_{(2,5)}]_3$		\vee		\vee

Работа системы производится аналогично Уровню1. Результаты:

$$\begin{aligned}
 [y_{(2,0)}]_7 &= [000]_2 = [000]_7 \\
 [y_{(2,1)}]_3 &= [010]_2 = [002]_3 \\
 [y_{(2,2)}]_4 &= [010]_2 = [002]_4 \\
 [y_{(2,3)}]_5 &= [111]_2 = [012]_5 \\
 [y_{(2,4)}]_2 &= [000]_2 \\
 [y_{(2,5)}]_3 &= [010]_2 = [002]_3
 \end{aligned}$$

Выходные значения $\{y_{2,j_2}\}$ должны быть сформированы в виде двух массивов:

$[\{y_{2,j_2}\}]_{k_{y_2,j_2}}$ и $[\{y_{2,j_2}\}]_2$. Это необходимо для проверки и контроля.

- **Уровень 3.** $i:=3$. Генерация: $l_3 = 2, \Rightarrow j_3 = 0, 1$. Затем система генерирует функции

$f_{(3,j_3)}$, связанные с ними числа y_{3,j_3} и основания k_{y_3,j_3} :

$\{y_{3,j_3}\}$	$y_{(3,0)}$	$y_{(3,1)}$
k_{y_3,j_3}	5	3

Генерация функций:

$\{y_{2,j_2}\}$	$[000]_7$	$[002]_3$	$[002]_4$	$[012]_5$	$[000]_2$	$[002]_3$
$[y_{(3,0)}]_5$	↓	↓			↓	
$[y_{(3,1)}]_3$						

Работа системы производится аналогично Уровню 2. Результаты:

- $[y_{(3,0)}]_5 = [101]_2 = [010]_5$
 $[y_{(3,1)}]_3 = [101]_2 = [012]_3$

Выходные значения $\{y_{3,j_3}\}$ должны быть сформированы в виде двух массивов:

$[\{y_{3,j_3}\}]_{k_{y_3,j_3}}$ и $[\{y_{3,j_3}\}]_2$. Это необходимо для проверки и контроля.

- **Уровень 4.** $i:=4$. Генерация: $l_4 = 1, \Rightarrow j_4 = 0$. Затем система генерирует функции $f_{(4,j_4)}$,

связанные с ними числа y_{4,j_4} и основания k_{y_4,j_4} :

$\{y_{4,j_4}\}$	$y_{(4,0)}$
k_{y_4,j_4}	4

Генерация функций:

$\{y_{3,j_3}\}$	$[010]_5$	$[012]_3$
$[y_{(4,0)}]_4$	✓	✓

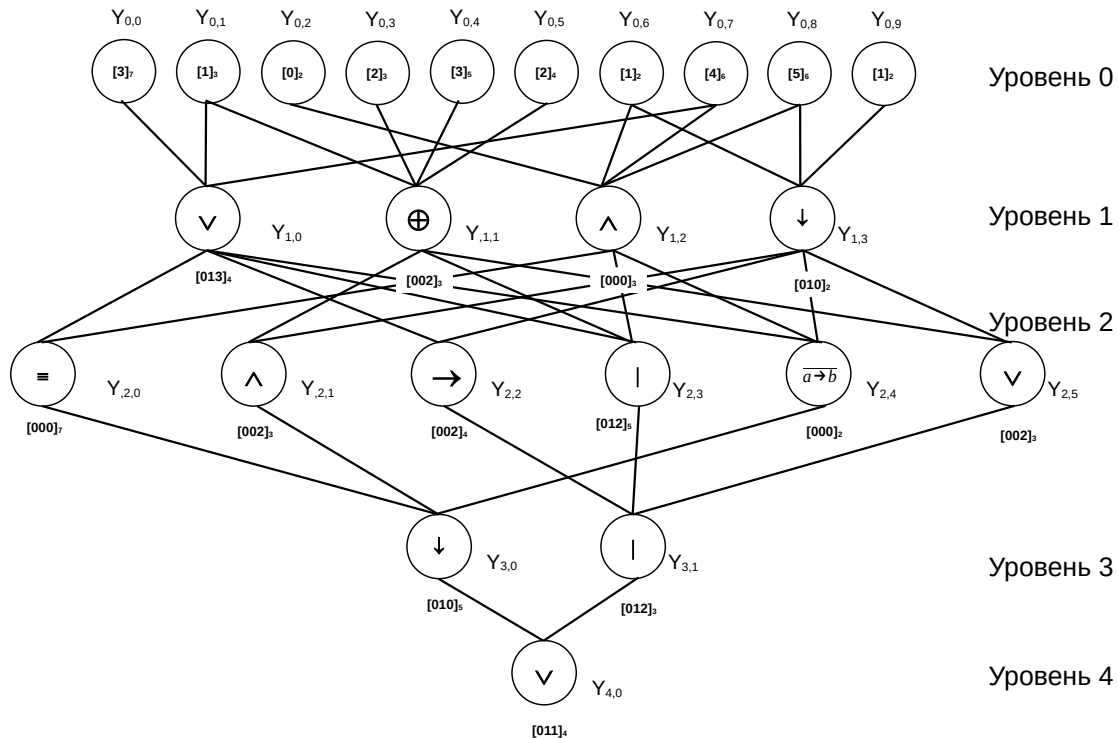
Работа системы производится аналогично Уровню 3. Результаты:

$$[y_{(4,0)}]_4 = [101]_2 = [011]_4$$

Таким образом, для исходных значений параметров модели система выдает $[y_{(4,0)}]_4 = [011]_4$ в качестве выходного параметра. После работы системы пользователь сам определяет выходное значение. В случае, если его значение совпадает с системой, на экран выдается "1", в противном случае "0".

Выходные значения $\{y_{4,j_4}\}$ должны быть сформированы в виде двух массивов: $[[y_{4,j_4}]_{k_{y_4,j_4}}]$ и $[[y_{4,j_4}]_2]$. Это необходимо для проверки и контроля.

Ярусно-параллельная форма алгоритма для данного примера имеет вид:



В результате генерации входных и настроечных параметров модели на экране монитора пользователя должны быть представлены следующие параметры:

- ✓ Количество уровней графа алгоритма m .
- ✓ Параметры $l_{m,i}$, $i = 0, 1, \dots, (m-1)$.
- ✓ Параметры $k_{y(i,l_i)}$.
- ✓ Массив входных чисел $\{[y(0, j_0)]_{k_{y(0,j_0)}}\}, j_0 \in [0, l_0 - 1]$
- ✓ Массив сгенерированных функций $f_{(i,j_i)}(y((i-1), j_{(i-1)})), f_{(i,j_i)} \in P_2$

Вся перечисленная информация должна предоставляться пользователю в виде многоконного интерфейса, алгоритм может быть представлен в виде граф-схемы либо совокупности таблиц уровней — см. материал ТЗ выше.

Ответ пользователя должен вводиться в отдельном окне.