10.7256/2305-6061.2013.01.1

§ 1 ПОКАЗАТЕЛИ КАЧЕСТВА И ПОВЫШЕНИЕ НАДЕЖНОСТИ ПРОГРАММНЫХ СИСТЕМ

В.С. Князьков, Т.В. Волченская

МАРКОВСКИЕ МОДЕЛИ ДЛЯ ОЦЕНКИ ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Аннотация. *В статье рассматривается иерархический подход к анализу сложности реализации параллельных вычислительных процессов. Вводятся понятия элементарных однородных и элементарных неоднородных вычислительных процессов, на основе которых строится анализ неоднородных ветвящихся параллельных и гетерогенных вычислительных процессов (ВП). Показано, что сложные параллельные и гетерогенные ВП представимы в виде дискретных Марковских процессов специального вида. Предложен подход, позволяющий преобразовывать такие процессы к классическим дискретным Марковским процессам с последовательной сменой состояний.*

Ключевые слова: *Программное обеспечение, эффективность, параллельные вычисления, вычислительные процессы, вычислительные структуры, марковские модели, организация вычислений, конвейерные вычисления, вычисления с параллелизмом объектов, оптимизация вычислений*

1. Организация и сложность неветвящихся процессов *SISD*-вычислений.

При использовании концепции SISD-вычислений ресурсы вычислительной среды (рис. 1) традиционно подразделяются на :

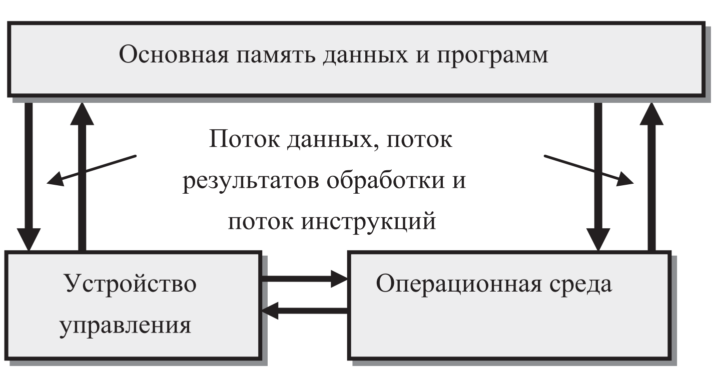

Рис. 1. Ресурсы SISD – вычислителя.

a) ресурсы, обеспечивающие управление вычислительными процессами (устройство управления) ;

b) ресурсы, обеспечивающие прием, размещение, хранение данных и генерацию потоков данных и команд в вычислительную среду (основная память данных и программ);

c) ресурсы, обеспечивающие преобразование и модификацию данных (операционная среда).

10.7256/2305-6061.2013.01.1

С точки зрения порядка использования ресурсов системы в процессе вычислений любой процесс SISD-вычислений представим последовательностью *M* элементарных подпроцессов, захватывающих и использующих ресурсы системы в определенной последовательности и в течении некоторого периода времени. Каждый элементарный подпроцесс в предельном случае – это результативно исполняемая инструкция программы.

В целом *SISD* – вычислитель можно рассматривать как физическую систему, в которой состояния ресурсов в общем случае изменяются случайным образом. Таким образом, вычислительный процесс, реализуемый в такой системе, можно рассматривать как случайный процесс, при выполнении которого в любой момент времени его вероятностные характеристики в будущем зависят только от текущего состояния системы и не зависят от того, в какой момент времени система перешла в это состояние. Если принять, что переходы системы из состояния в состояние при вычислениях происходят мгновенно и все возможные состояния системы определены, то такой процесс есть дискретный Марковским процесс и для его описания и анализа соответственно применимы классические положения теории Марковский процессов.

Можно выделить два основных типа процессов *SISD*-вычислений – однородные и неоднородные вычислительные процессы. При однородных вычислениях сложность исполнения каждого из элементарных подпроцессов примерно одинакова, то есть затрачиваются равные временные и аппаратные ресурсы. При неоднородных вычислениях – затраты ресурсов существенно различны. С другой стороны, любой неоднородный процесс представим определенной последовательностью однородных процессов. Таким образом, эффективность организации выполнения однородных процессов в системе играет ключевую роль при проектировании сложных гетерогенных вычислительных процессов. В связи с этим сначала рассмотрим особенности организации именно однородных вычислительных процессов, реализуемых на базе *SISD*-вычислителя с классической организацией.

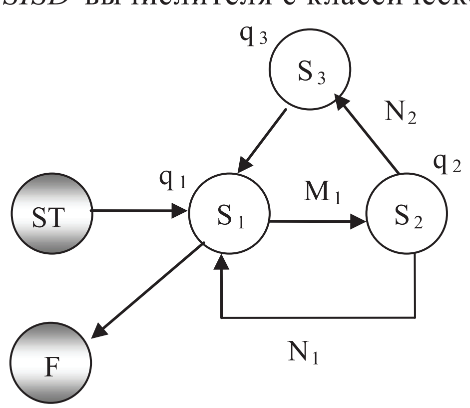

Рис. 2.
Модель организации процесса однородных SISD– вычислений.

Процесс однородных SISD-вычислений представим дискретным Марковским процессом (рис. 2) со следующими состояниями:

$ST(t_0)$ – состояние, в котором находится SISD-вычислитель до запуска вычислительного процесса ; $S_1(t_i)$ – состояние активной работы ресурсов *SISD*-вычислителя в t_i момент времени для выполнения подпроцессов выборки данных и команд из основной памяти и их ввод на исполнение в операционные ресурсы; $S_2(t_i)$ – состояние, соответствующее активной работе операционных ресурсов *SISD*-вычислителя в t_i момент времени; $S_3(t_i)$ – состояние, соответствующее активной ра-

10.7256/2305-6061.2013.01.1

боте ресурсов *SISD*-вычислителя в t_i момент времени для обеспечения вывода результатов вычислений из операционной среды и их ввода в основную память; $F(t_i)$ – финальное состояние, соответствующее полному завершению вычислительного процесса и его останов.

Для рассматриваемого случая N_1 – число элементарных подпроцессов, при исполнении которых требуется использовать как операционные ресурсы так и ресурсы памяти *SISD*-вычислителя. N_2 – число элементарных подпроцессов, при исполнении которых используются только операционные ресурсы и не используются ресурсы памяти. q_i – сложность реализации элементарного подпроцесса в $S_i(t_i)$ состоянии.

Рассматриваемая модель однородных *SISD*-вычислительных процессов, во-первых, определяет порядок порождения состояний процесса и их сложность. Во-вторых, отображает вычислительный процесс как случайный процесс, что позволяет учитывать бесконечное число реализаций реального вычислительного процесса, определяемых как алгоритмом реализации, так и исходными данными.

Эффективность процессов однородных вычислений целесообразно оценивать с точки зрения сложности их реализации – чем меньше сложность реализации процесса, тем процесс более эффективно организован. В качестве меры сложности в рамках данной работы рассматривается временная сложность, оцениваемая количеством шагов вычислений, необходимых для однократного и результативно выполнения вычислительного процесса. Причем, причем шаг вычислений – выполнение вычислительного подпроцесса в соответствующем состоянии системы.

Так как при однородных вычислениях состав элементарных подпроцессов неизменен, то вероятности переходов $P_{i,j}(t)$ вычислительного процесса из состояния $S_i(t)$ в состояние $S_j(t+1)$ не изменяются и определяются значениями N_1 и N_2:

$$\begin{aligned}
&P_{0,1} = P_{3,1} = 1; \\
&P_{1,2} = M/(M+1); \\
&P_{2,3} = N_2/(N_1+N_2); \\
&P_{2,1} = N_1/(N_1+N_2); \\
&P_{2,k} = 1/(M+1).
\end{aligned} \tag{1.1}$$

При однократной и результативной реализации процесса среднее число попаданий процесса (d_i^{cp}) в состояния $S_1(t_i)$, $S_2(t_i)$, $S_3(t_i)$, $S_k(t_i)$ очевидно определяются следующим образом :

$$\begin{aligned}
&d_1^{cp} = d_0 \times P_{0,1} + d_2^{cp} \times P_{2,1} + d_3^{cp} \times P_{3,1} \\
&d_2^{cp} = d_1^{cp} \times P_{1,2} \\
&d_3^{cp} = d_2^{cp} \times P_{2,3} \\
&d_k = d_2^{cp} \times P_{2,k}
\end{aligned} \tag{1.2}$$

10.7256/2305-6061.2013.01.1

При $d_0 = 1$ и значениях $P_{i,j}(t)$ из 1.1. :

$$
\begin{aligned}
d_0 &= 1 \\
d_1^{cp} &= (M + 1) \\
d_2^{cp} &= (M) \\
d_3^{cp} &= M N_2 / (N_1 + N_2) \\
d_k &= 1.
\end{aligned}
\qquad (1.3)
$$

При выборе в качестве критерия критерий средней временной сложности Qпроцессаср (t) однократной и результативной реализации процесса достаточно определиться со значениями временной сложности q_0, q_1, q_2, q_3, q_k выполнения каждого из подпроцессов в соответствующем состоянии. Тогда

$$Q^{процесса}{}_{cp}(t) = q_1 d_1^{cp} + q_2 d_2^{cp} + q_3 d_3^{cp}. \qquad (1.4)$$

Подставляя значения d_i^{cp} и выполнив несложные преобразования получим, что средняя временная сложность выполнения однородного вычислительного процесса в SIMD-вычислителе будет определяться выражением:

$$Q^{процесса}_{cp} = \frac{(N_1 + N_2)(M+1)q_1 + M(N_1 + N_2)q_2 + MN_2 q_3}{N_1 + N_2}$$

При условии, что $q_1 = q_2 = q_3 = q_э$

$$Q^{процесса}_{cp} = 2M + \frac{MN_2}{N_1 + N_2} \qquad (1.5)$$

В частности, при выполнении однородных вычислений без записи результатов в основную память ($N_2 = 0$ – типичная ситуация для вычислителей на основе концепции RISC вычислений) $N_1 = M$ и

$$Q^{процесса}{}_{cp}(t) = 2M + 1. \qquad (1.6)$$

При выполнении на каждом шаге вычислений как операций обработки данных, так и записи результатов их выполнения в память $N_1 = N_2 = M$ (типичная ситуация для *CISC* вычислений) средняя временная сложность реализации вычислительного процесса будет определяться выражением

$$Q^{процесса}{}_{cp}(t) = (3M + 1). \qquad (1.7)$$

Отсюда имеем, что при больших значениях $M >> 1$ *RISC*-вычисления с точки зрения их временной сложности в 1,5 раза эффективнее, чем *CISC*-вычисления.

10.7256/2305-6061.2013.01.1

При выполнении неоднородных вычислений элементарные подпроцессы существенно различны по сложности. Например, в период времени *t...(t+10)* будет выполнено четыре подпроцесса типа k_1 со сложностью q_1 , три подпроцесса типа k_2 со сложностью q_2 и три подпроцесса типа k_3 со сложностью q_3:

$$\ldots k_1(t), k_2(t+1), k_2(t+2), k_3(t+3), k_1(t+4), k_2(t+5), k_1(t+6), k_3(t+7), k_3(t+8), k_1(t+9), k_1(t+10)\ldots$$

Таким образом, неоднородный процесс состоит из некоторой совокупности однородных подпроцессов, сложность реализации которых определяется по выше рассмотренной методике.

Пусть *X-процесс* – однородный случайный вычислительный процесс, для которого справедливы рассмотренные выше положения. В этом случае процесс неоднородных вычислений в среде *SISD*-вычислителя представим Марковским графом, показанном на рис. 3, где :

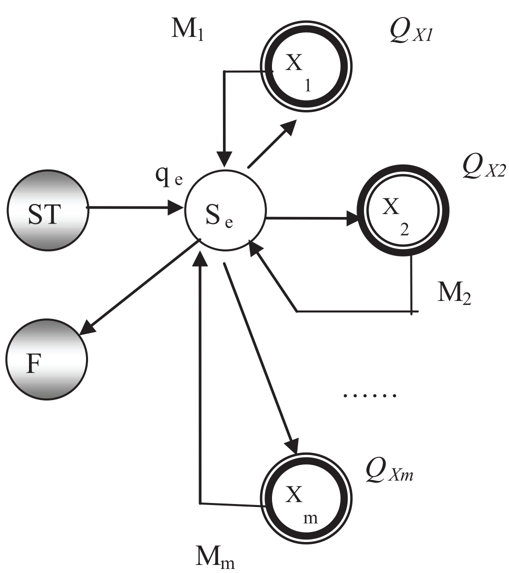

Рис. 3. Модель организации процесса неоднородных SISD– вычислений.

$ST(t_0)$ – исходное состояние всех ресурсов *SISD*-вычислителя и старт вычислительного процесса;

X_i – состояние, отображающее выполнение i-го подпроцесса однородных *SISD*-вычислений сложности $\boldsymbol{Q_{X1}}$**,** содержащий поток из M_i однотипных элементарных подпроцессов сложности q_i ;

S_e – состояние, отображающее переключение вычислительного процесса на обработку следующего потока однотипных инструкций со сложностью q_e ;

F – финальное состояние – завершение вычислительного процесса.

При таком представлении процесса неоднородных вычислений его сложность $Q^{процесса}{}_{ср}(t)$ определяется суммарной сложностью *X*-подпроцессов, определяемой по выражения 1.1-1.7 при $M = M_1 + M \ldots M_R$:

$$Q^{процесса}{}_{ср}(t) = \sum_{i=1}^{R} \frac{M_i}{M+1} Q_{X\,i.,} \tag{1.8}$$

10.7256/2305-6061.2013.01.1

2. Организация и сложность ветвящихся процессов *SISD*-вычислений.

Ранее были рассмотрены однородные и неоднородные процессы *SISD*-вычислений, при выполнении которых не предусматривались ветвления – альтернативные направления развития процесса в зависимости от выполнения или невыполнения некоторого условия отсутствовали. В случае ветвящихся однородных, неоднородных и составных *SISD*-процессов их сложности можно достаточно просто определить на основе ранее проведенных рассуждений исходя из следующего.

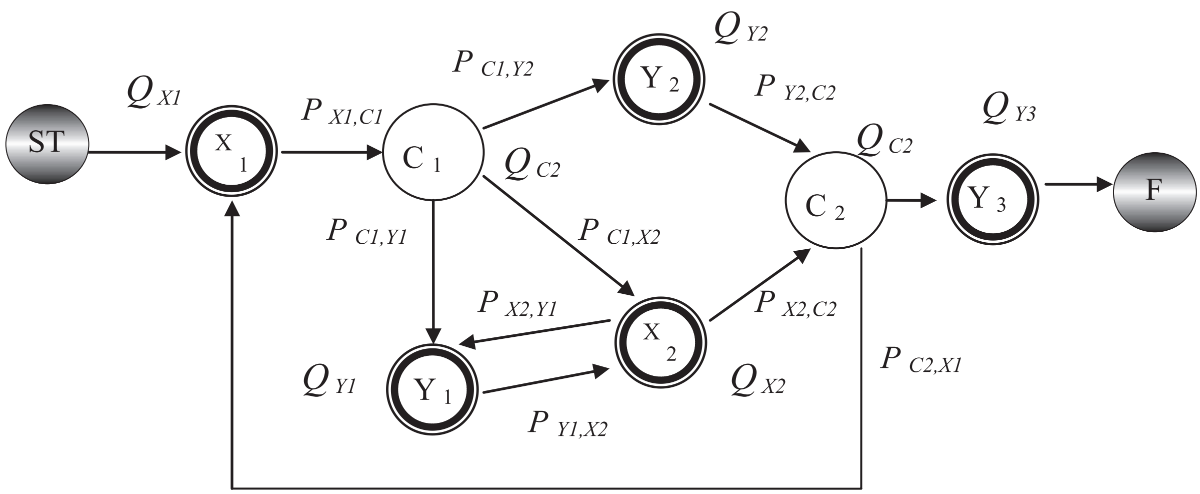

Рис. 4. Модель организации процесса неоднородных SISD– вычислений.

Пусть Y_i – процесс неоднородных, а X_i – процесс однородных *SISD*-вычислений, $C_i(t)$ – состояния вычислительного процесса, в котором принимается решение о продолжении процесса вычислений по одной из альтернативных ветвей в зависимости от значения некоторой функции условия разветвления $F_i(t)$. В этом случае ветвящийся процесс в общем виде представим графом Маркова, включающим $C_i(t_j)$ вершины, $X_i(t_j)$ –вершины, $Y_i(t_j)$ – вершины, начальную вершину *ST* и финальную вершину *F*.

Пример ветвящегося процесса *SISD*-вычислений в форме граф Маркова показан на рис. 4., где $P_{Xi,Ci}$ – вероятности перехода процесса из состояния однородных вычислений $X_i(t)$ со сложностью Q_{Xi} в состояние анализа условия ветвления $C_i(t)$ со сложностью Q_{CI}. Аналогичный смысл имеют и остальные обозначения, приведенные на рис. 4.

Оценку сложности ветвящихся процессов *SISD*-вычислений составного типа на основе рассмотренных выше положений можно выполнить следующим образом:

1. определить сложность выполнения процесса Q_{Xi} и Q_{Yi}, соответствующих сложности выполнения X-процессов и Y-процессов;

10.7256/2305-6061.2013.01.1

2. определить сложность Q_{Ci} выполнения состояний C_i составного процесса;

3. определить число попаданий D_i в каждое из состояний вычислительного процесса при его результативной и однократной реализации путем решения следующей системы линейных уравнений

$$
\begin{aligned}
&(p_{1,1} -1)D_1 + (p_{2,1})D_2 + (p_{3,1})D_3 + \ldots (p_{k-2,1})D_{k-2} + (p_{k-1,1})D_{k-1} = -1;\\
&(p_{1,2})D_1 + (p_{2,2} -1)D_2 + (p_{3,2})D_3 + \ldots (p_{k-2,2})D_{k-2} + (p_{k-1,2})D_{k-1} = 0;\\
&(p_{1,3})D_1 + (p_{2,3})D_2 + (p_{3,3} -1)D_3 + \ldots (p_{k-2,3})D_{k-2} + (p_{k-1,3})D_{k-1} = 0;\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&(p_{1,k})D_1 + (p_{2,k})D_2 + (p_{3,k})D_3 + \ldots (p_{k-2,K})D_{k-2} + (p_{k-1,1-1})D_{k-1} = 0.
\end{aligned}
\qquad (2.1.)
$$

4. определить сложность выполнения процесса по выражению 2.2, где R – количество состояний процесса :

$$Q_{cp}(t) = \sum_{i=1}^{R} (D_{Xi} Q_{Xi.} + D_{Yi} Q_{Yi.} + D_{Ci} Q_{Ci.}) \qquad (2.2)$$

3. Организация и сложность процессов конвейерных MISD-вычислений.

При использовании концепции конвейерных вычислений ресурсы *MISD*– вычислителя подразделяется на группы ресурсов, аналогичные ресурсам *SISD*– вычислителя (рис. 5). Разница заключается только в том, что в группе операционных ресурсов (магистрали обработки данных) обеспечивается параллельное исполнение нескольких взаимосвязанных по данным элементарных $X_i(t)$-процессов. Результаты вычислений каждого $X_i(t)$-процесса в *i*-ом сегменте являются входными данными для $X_{i+1}(t+1)$-процесса в $(i+1)$ -ом сегменте.

Однородный неветвящийся процесс *MISD*-вычислений (*Z-процесс*) в этом случае представим Марковским процессом из $(2k-1)$ состояний (рис. 6), где : $S_1(t)$, $S_2(t)$,..., $S_k(t)$ состояния, соответствующие этапам запуска вычислений в сегментах операционной магистрали конвейера, $S_k(t)$ – состояние параллельных вычислений во всех *k* сегментах магистрали, $S_{(k+1)}(t)$, $S_{(k+2)}(t)$, ..., $S_{(2k-1)}(t)$ – состояния, соответствующие этапам завершения вычислений в сегментах магистрали конвейера, *ST* начальное и *F* финальное состояния процесса. *M* – ко-

10.7256/2305-6061.2013.01.1

личество элементов обрабатываемого потока данных, q_i – сложность реализации $S_i(t)$ состояния процесса, k – количество сегментов магистрали обработки данных.

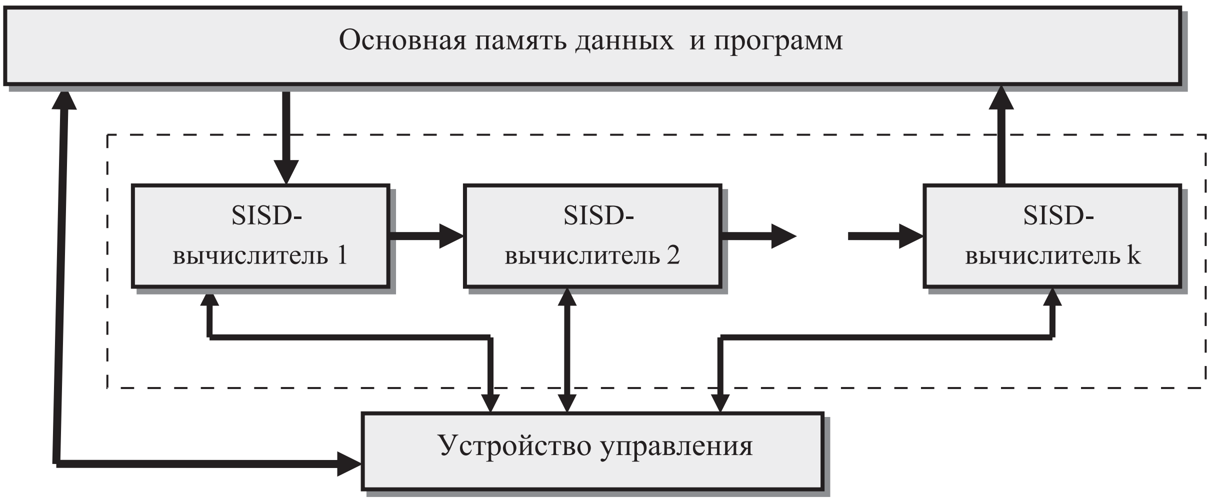

Рис. 5.
Ресурсы MISD-вычислителя.

Так как сложность реализации вычислений в сегментах магистрали примерно одинакова, то $q_1 = q_2 = q_3 = ... = q_{ст}$ и сложность реализации процесса *MISD*-вычислений в операционной магистрали конвейера определяется как

$$Q^{процесса}_{ср}(t) = (M - 1 + k)\, q_{ст} .$$

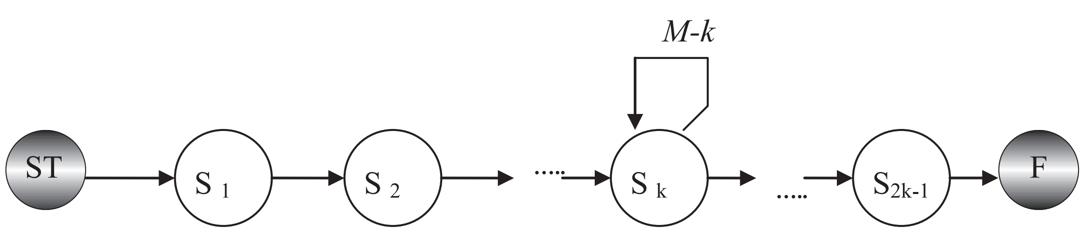

Рис. 6.
Модель процесса MISD– вычислений

Сложность неоднородных ветвящихся *MISD*-процессов определяется аналогично сложности ветвящихся *SISD*-процессов после предварительной оценки сложности входящих в его состав неветвящихся Z_i– процессов. Модель организации неоднородного ветвящегося *MISD*-процесса показан на рис. 7, где : $P_{Zi,Ci}$ –

10.7256/2305-6061.2013.01.1

вероятности перехода процесса из состояния $Z_i(t)$ однородных *MISD*-вычислений в состояние $C_i\,(t)$ – состояние анализа условия ветвления неоднородного ветвящегося *MISD*-процесса; Q_{Zi} – сложность реализации $Z_i(t)$ однородного процесса *MISD*-вычислений; Q_{Ci} – сложность реализации $C_i\,(t)$ состояния.

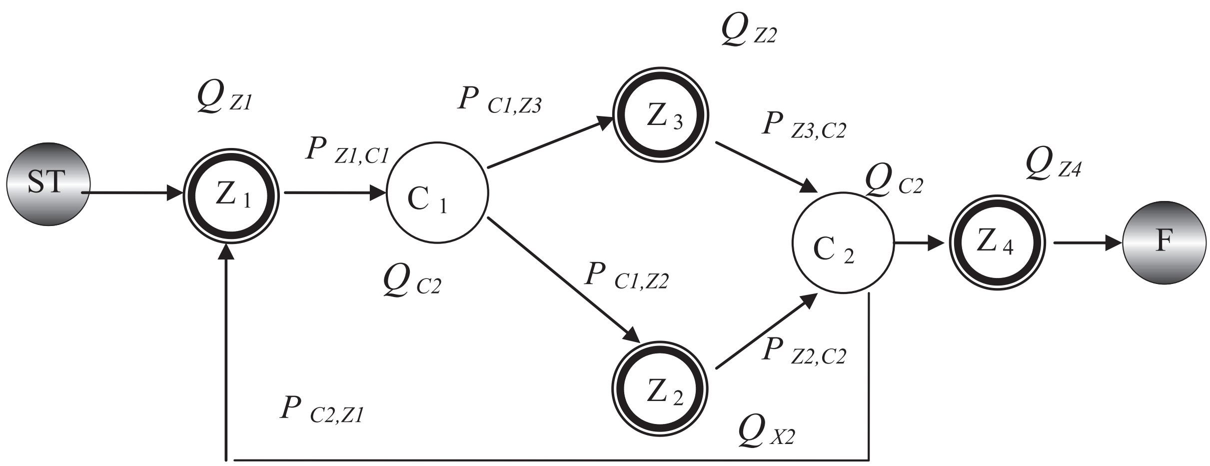

Рис. 7.
Модель организации неоднородных MISD вычислений.

В рассматриваемом примере вычисление функций переходов (состояния $C_i\,(t)$) реализуется управляющими, а исполнение $Z_i(t)$ однородных процессов – операционными ресурсами *MISD*-вычислителя.

В общем случае *Z*– процессы, входящие в качестве подпроцессов в неоднородный ветвящийся процесс *MISD*-вычислений, различны по типу и соответственно значения Q_{Zi} будут существенно отличаться. В этом случае сложность процесса определяется по методике раздела 2 :

1. Определить сложность выполнения отдельных Z-процессов ;
2. Определить сложность выполнения состояний C_i ;
3. Определить число попаданий D_i ветвящегося неоднородного *MISD*-процесса в каждое из состояний при его результативной и однократной реализации по системе 2.1.;
4. Определить сложность выполнения ветвящегося неоднородного *MISD*-процесса по выражению 2.2.

3. Организация и сложность процессов SIMD-вычислений.

При использовании концепции *SIMD*-вычислений операционные ресурсы системы представляются группировкой параллельного работающих секций

SISD-вычислителей, в которых одновременно реализуются однотипные *X*– или *Y*-процессы (рис. 8).

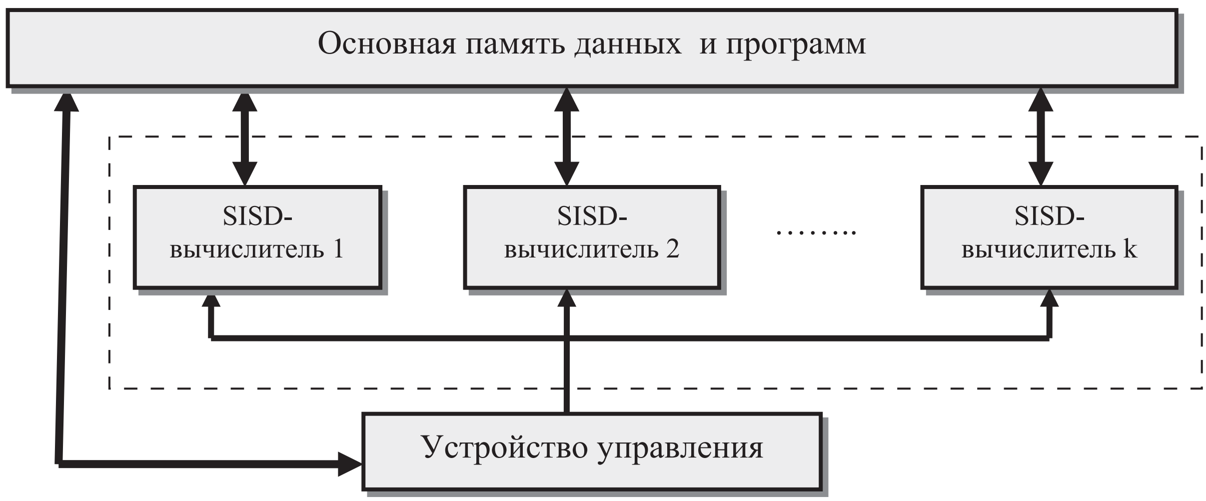

Рис. 8. Ресурсы SIMD-вычислителя.

Такой тип организации параллельных неветвящихся процессов *SIMD*-вычислений (*W (t) – процесс*) представим в виде Марковского процесса со следующими состояниями (рис. 9), где :

- *ST* начальное и *F* финальное состояния процесса *SIMD*-вычислений ;
- $Y_{j,i}$ *(t)– j –ый* процесс неоднородных *SISD*– вычислений сложности $Q_{j,i}$, исполняемый в *i-ой* секции *SIMD*-вычислителя;
- $ST_{(-)SIMD}$ *(t)* – состояние, определяющее загрузку группы однотипных $Y_{j,i}$ – процессов в *SISD*-секции *SIMD*-вычислителя и запуск параллельных вычислений;
- $E_{(+)SIMD}$ *(t)* – состояние, определяющее завершение всех $Y_{j,i}$ – процессов во всех *SISD*-секциях и останов параллельных вычислений;
- $q_{(-)}$ – сложность реализации состояния $ST_{(-)SIMD}$ *(t)* ;
- $q_{(-)}$ – сложность реализации состояния $E_{(+)SIMD}$ *(t) ;*
- $N_{k\ (+)}$ – число повторений реализации W_k *(t)*–процесса при выполнении вычислений.

Очевидно, что в этом случае сложность параллельного вычислительного процесса в SISD-секциях определяется как *max* $(Q_{j,I})$, где *i =1,2,…,m*. Тогда сложность $Q_{Wi(t)}$ реализации всего $W_i(t)$–процесса соответственно будет равна

$$Q_{Wi(t)} = max\ (Q_{j,I}) + q_{(-)} + q_{(+)};\ i = 1,2,\ldots,m. \qquad (3.1.)$$

10.7256/2305-6061.2013.01.1

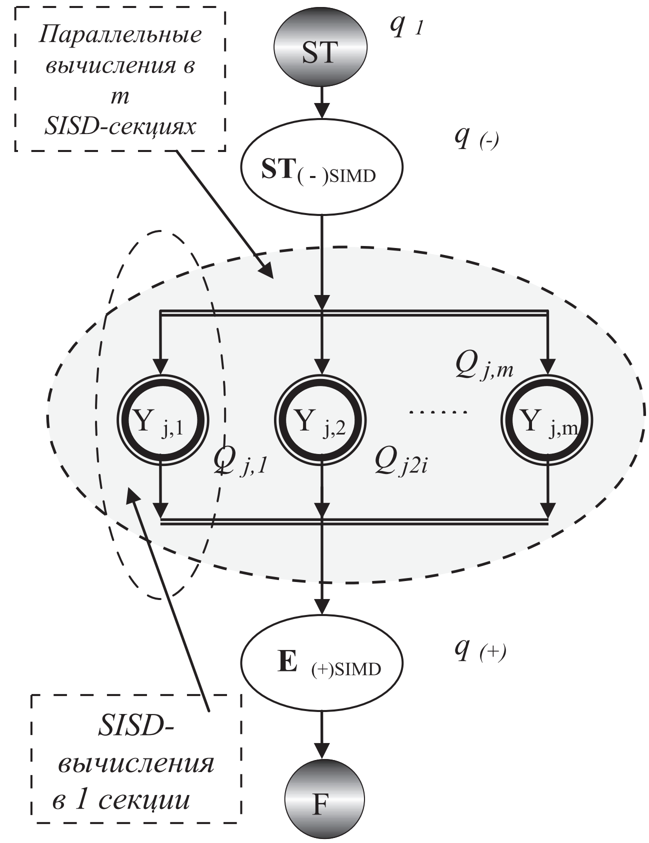

Рис.7.
Модель организации SIMD– вычислений

При анализе сложности реализации неоднородных ветвящихся процессов *SIMD*-вычислений очевидно достаточно представить такой процесс как совокупность $W_i(t)$–процессов, взаимодействующих между собой через $C_i\ (t)$ состояния, в которых выполняется анализ условий запуска альтернативных ветвей $W_i(t)$–процессов. То есть механизм анализа в этом случае полностью аналогичен механизму, приведенному выше для неоднородных ветвящихся *MISD*-процессов. Разница заключается только в том, что определение сложности отдельных $W_i(t)$–процессов необходимо выполнить соответственно с использованием выражения 3.1.

Вариант организации неоднородного ветвящегося процесса *SIMD*-вычислений показан на рис. 8. В этом случае для выполнения анализа и оценки сложности реализации процесса в целом достаточно выполнить подстановки $W_i(t)$–процессов с соответствующей им сложностью вычислений в альтернативные ветви. В итоге модель процесса сводится к рассмотренной ранее модели неоднородных ветвящихся псевдо SISD-процессов, где в качестве состояний определены состояния исполнения неветвящихся $W_i(t)$–процессов (рис. 9), и далее выполняется оценка сложности по методике оценки неоднородных ветвящихся SISD-процессов.

10.7256/2305-6061.2013.01.1

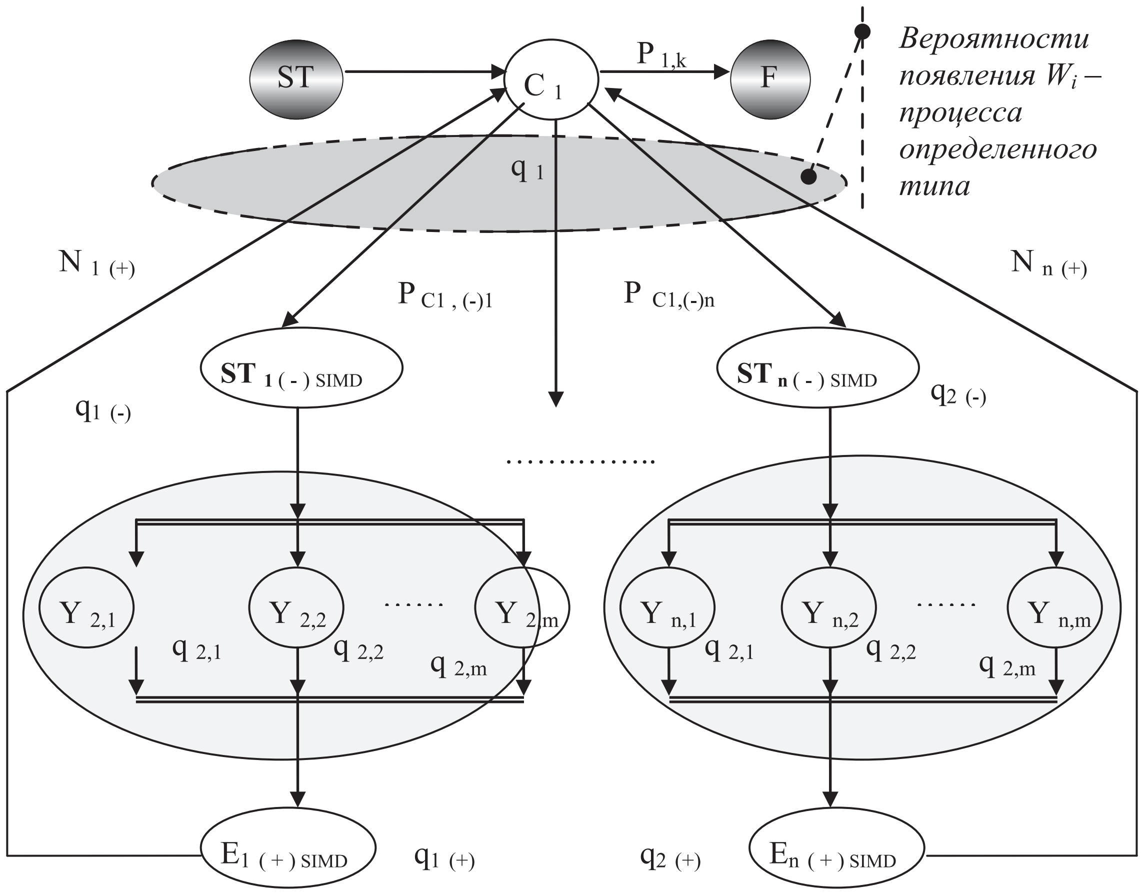

Рис. 8.
Модель организации ветвящегося процесса SIMD-вычислений.

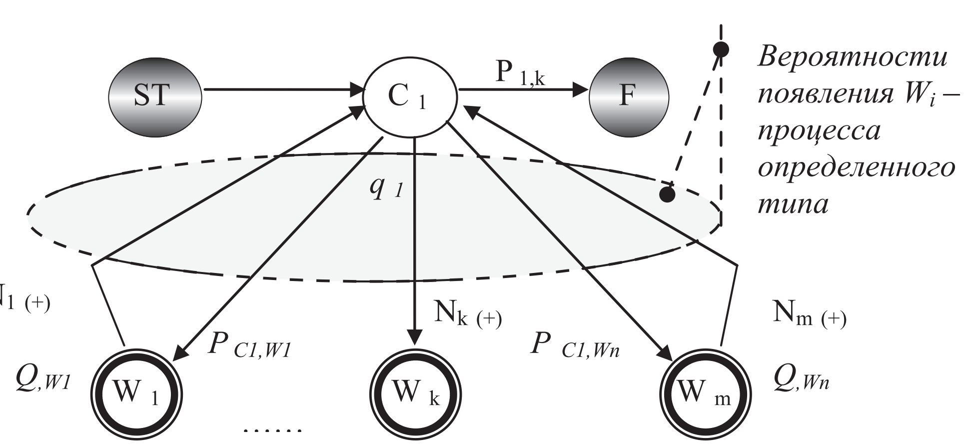

Рис. 9.
Модель ветвящегося процесса SIMD-вычислений как неоднородного ветвящегося псевдо SIMD-процесса.

10.7256/2305-6061.2013.01.1

4. Организация и сложность процессов MIMD-вычислений.

При использовании концепции *MIMD*-вычислений операционные ресурсы системы классически представлены группировкой параллельного и независимо работающих секций *SISD*-вычислителей (рис. 10), в которых одновременно выполняются разнотипные *X*– или *Y*-процессы в совокупности составляющие *G-процесс*. Управление вычислительным процессом в каждой из секций индивидуальное и определяется алгоритмом исполнения процесса.

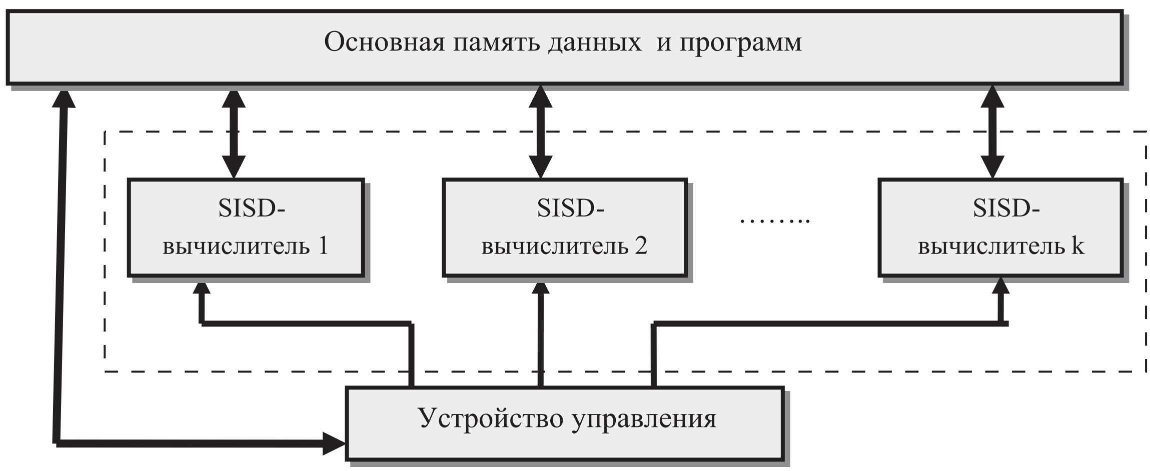

Рис. 10.
Ресурсы MIMD-вычислителя.

Вариант организации синхронного *G-процесса MIMD*-вычислений в виде дискретного графа Маркова приведен на рис. 11., где : $E^{S}_{(+)\,MIMD}(t)$ –состояние останова параллельного вычислительного процесса MIMD– вычислений при выполнении условия синхронизации – полного завершения всех параллельно исполняемых неоднородных ветвящихся SISD-процессов в параллельно работающих вычислительных секциях; $E_i\,(t)$ – состояние завершения *i,j – го* неоднородного ветвящегося SISD-процесса; $ST_{(-)\,MIMD\,(t)}$ –синхронный старт SISD-процессов в параллельно работающие вычислительные секции; $M_{i,j}$ – число повторений ветвей при выполнении соответствующих SISD-процессов при их однократной реализации ; $N^S(+)$ – число повторений параллельного исполнения неоднородных ветвящихся SISD-процессов в режиме синхронных MIMD– вычислений; $Q_{i,j}$ – сложность реализации *i,j – го* неоднородного ветвящегося SISD-процесса; C_1 – условие завершения вычислений, $P_{C1,\,ST(-)}$ – вероятность старта, а $P_{C1,F}$ – вероятность завершения параллельно исполняемых вычислительных процессов.

10.7256/2305-6061.2013.01.1

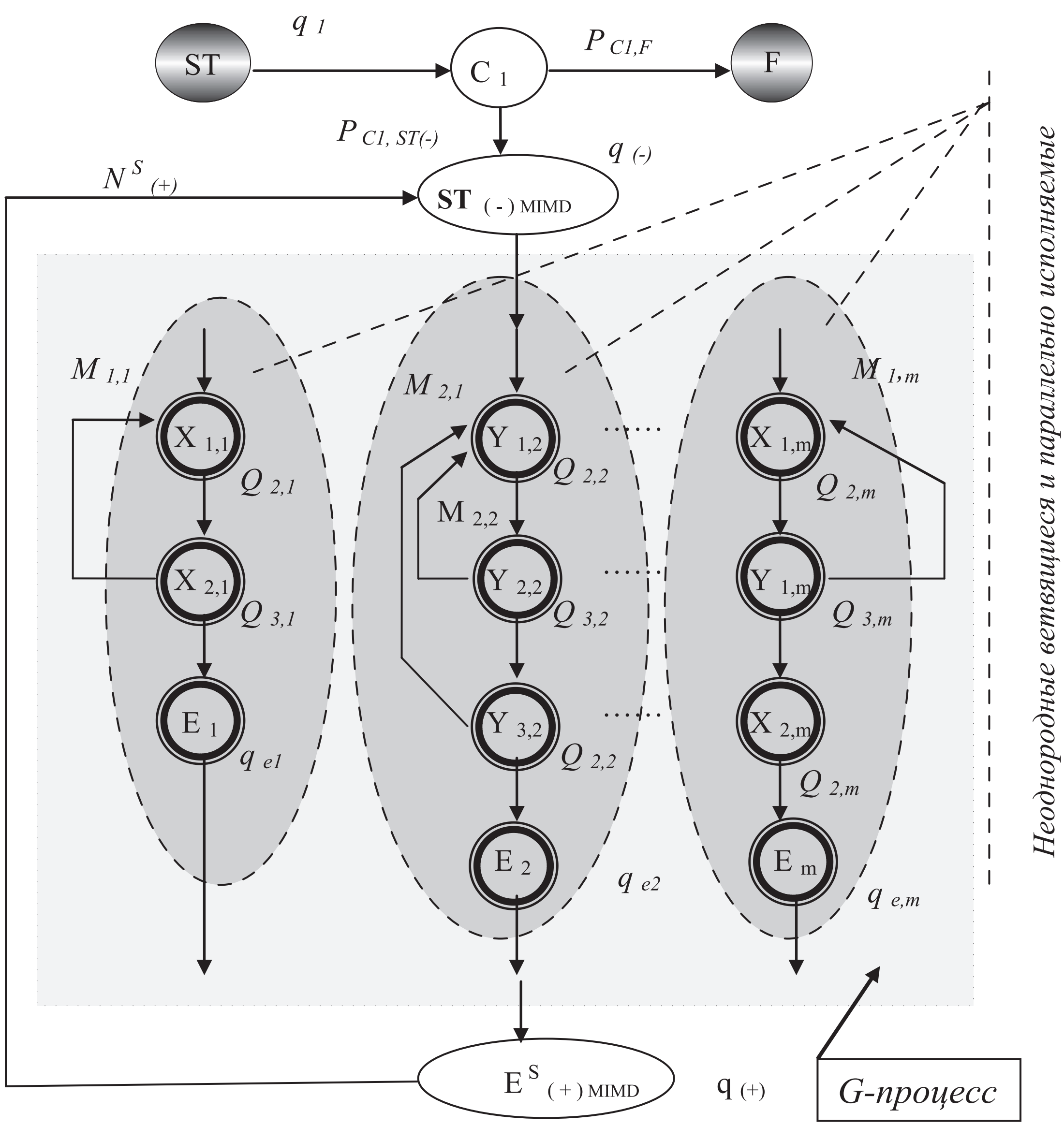

Рис. 11.
Модель организации синхронных MIMD– вычислений.

В случае выполнения процессов синхронных *MIMD*-вычислений процесс попадает в состояние $E^{S}_{(+)\,MIMD}$ *(t)* при условии завершения всех параллельно исполняемых процессов, то есть переход всех параллельно исполняемых процессов в состояния E_1 *(t)*, E_2 *(t,)* ..... E_m *(t): состояние* $E^{S}_{(+)MIMD}$ *(t) «истинно» если E1 (t)& E2 (t)& ..... & Em (t)– «истинно».*

Отсюда сложность однократной реализации этапа *MIMD*-вычислений определятся наиболее сложным из параллельно исполняемых SISD-процессов *G-процесса* . Таким образом, модель процесса *MIMD*-вычислений представима дискретным Марковским

10.7256/2305-6061.2013.01.1

процессом, в котором состояния *MIMD*-вычислений интерпретируются состояниями исполнения различных по сложности *G-процессов*.

При таком подходе для рассмотренного на рис.11 варианта вычислений вычислительный процесс представим графом дискретного Марковского процесса, приведенном на рис. 12.

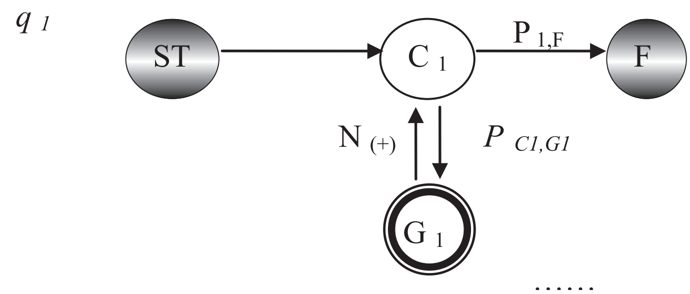

Рис. 12.
Модель процесса MIMD вычислений на основе G-процессов.

В заключение следует отметить, что рассмотренный подход к оценке эффективности организации параллельных вычислений позволяет достаточно просто рассматривать сложные гетерогенные вычислительные процессы, при исполнении которых выполняются процессы однородных и неоднородных последовательных вычислений, вычислений в конвейерном режиме и вычисления с параллелизмом объектов, вычисления с параллелизмом ветвей и естественным параллелизмом в различных сочетаниях. Последовательная иерархическая свертка гетерогенного вычислительного процесса к псевдо последовательному процессу позволяет выявить его структуру, определиться со сложностью реализации его составных частей и в итоге определиться со стратегией его оптимизации с точки зрения выбранного критерия сложности.

Список литературы:

1. Основы теории вычислительных систем. Под редакцией Майорова С.А. М., «Высшая школа», 1978, 407 с.
2. *Князьков В.С., Волченская Т.В.* Конвейерные вычисления и структуры: формальные модели и аналитические оценки сложности вычислений. В кн.: Материалы 12 Международной научно-технической конференции «Проблемы передачи и обработки информации в сетях и системах телекоммуникаций», Рязань, 2004, с. 128-131.
3. *Князьков В.С., Потапов А.А.* Методика оценки трудоемкости реализации матричных мультипроцессорных систем. Труды международного симпозиума «Актуальные проблемы науки и образования», Пенза, 2003, с. 400-402.

10.7256/2305-6061.2013.01.1

4. *Князьков В.С.* Способы организации и сложность массивных вычислений в одномерных итеративно-битовых процессорных средах // Вычислительная техника в автоматизированных системах контроля и управления: Межвузовский сборник научных трудов – Пенза: Издательство Пензенского государственного университета, 1999 г.– Вып. 26.– с. 3-9.
5. *Князьков В.С.* Арифметико-конвейерные вычисления в двумерных однородных средах // Материалы 3 Международной научно-технической конференции «Новые информационные технологии и системы» 10-11 декабря 1998 г., г. Пенза, – Пенза,1998, С. 53-54.
6. *Князьков В.С.* Общая оценка сложности реализации массивно-клеточных вычислений в итерационно-битовых вычислительных структурах с многомерной организацией. //Известия ТРТУ, N 3,1997,с.218.
7. *Князьков В.С.* Двумерные итеративно-битовые процессоры: временная и пространственная сложность последовательно-массивных итерационно-разрядных вычислений //Материалы 2 -ой Междунар. н/т конф. «Новые информационные технологии и системы,ч.1,1996, Пенза,С.108-109.
8. *Князьков В.С., Бикташев Р.А.* Архитектура параллельных вычислительных систем.-Пенза,1993.-с.166.
9. *Князьков В, С., Волченская Т. В.* Способы построения конвейерных вычислительных структур с управлением коммутации потоков данных. Деп. рук., ВИНИТИ, N 5581В9Д от 31. 10. 90, Минвуз РСФСР, Пенза, Пензенский политехнический и – нт, 1990, 10 с.

Библиография:

1. Основы теории вычислительных систем. Под редакцией Майорова С.А. М., «Высшая школа», 1978, 407 с.
2. Князьков В.С., Волченская Т.В. Конвейерные вычисления и структуры: формальные модели и аналитические оценки сложности вычислений. В кн.: Материалы 12 Международной научно-технической конференции «Проблемы передачи и обработки информации в сетях и системах телекоммуникаций», Рязань, 2004, с. 128-131.
3. Князьков В.С., Потапов А.А. Методика оценки трудоемкости реализации матричных мультипроцессорных систем. Труды международного симпозиума «Актуальные проблемы науки и образования», Пенза, 2003, с. 400-402.
4. Князьков В.С. Способы организации и сложность массивных вычислений в одномерных итеративно-битовых процессорных средах // Вычислительная техника в автоматизированных системах контроля и управления: Межвузовский сборник научных трудов – Пенза: Издательство Пензенского государственного университета, 1999 г.– Вып. 26.– с. 3-9.
5. Князьков В.С. Арифметико-конвейерные вычисления в двумерных однородных средах // Материалы 3 Международной научно-технической конференции

10.7256/2305-6061.2013.01.1

“Новые информационные технологии и системы” 10-11 декабря 1998 г., г. Пенза, – Пенза,1998, С. 53-54.
6. Князьков В.С. Общая оценка сложности реализации массивно-клеточных вычислений в итерационно-битовых вычислительных структурах с многомерной организацией. //Известия ТРТУ, N 3,1997,с.218.
7. Князьков В.С. Двумерные итеративно-битовые процессоры: временная и пространственная сложность последовательно-массивных итерационно-разрядных вычислений //Материалы 2-ой Междунар. н/т конф. “Новые информационные технологии и системы,ч.1,1996, Пенза,С.108-109.
8. Князьков В.С., Бикташев Р.А. Архитектура параллельных вычислительных систем.-Пенза,1993.-с.166.
9. Князьков В, С., Волченская Т. В. Способы построения конвейерных вычислительных структур с управлением коммутации потоков данных. Деп. рук., ВИНИТИ, N 5581В9Д от 31. 10. 90, Минвуз РСФСР, Пенза, Пензенский политехнический и – нт, 1990, 10 с.

References (transliteration):

1. Osnovy teorii vychislitel’nykh sistem. Pod redaktsiey Mayorova S.A. M., «Vysshaya shkola», 1978, 407 s.
2. Knyaz’kov V.S., Volchenskaya T.V. Konveyernye vychisleniya i struktury: formal’nye modeli i analiticheskie otsenki slozhnosti vychisleniy. V kn.: Materialy 12 Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii «Problemy peredachi i obrabotki informatsii v setyakh i sistemakh telekommunikatsiy», Ryazan’, 2004, s. 128-131.
3. Knyaz’kov V.S., Potapov A.A. Metodika otsenki trudoemkosti realizatsii matrichnykh mul’tiprotsessornykh sistem. Trudy mezhdunarodnogo simpoziuma «Aktual’nye problemy nauki i obrazovaniya», Penza, 2003, s. 400-402.
4. Knyaz’kov V.S. Sposoby organizatsii i slozhnost’ massivnykh vychisleniy v odnomernykh iterativno-bitovykh protsessornykh sredakh // Vychislitel’naya tekhnika v avtomatizirovannykh sistemakh kontrolya i upravleniya: Mezhvuzovskiy sbornik nauchnykh trudov – Penza: Izdatel’stvo Penzenskogo gosudarstvennogo universiteta, 1999 g.– Vyp. 26.– s. 3-9.
5. Knyaz’kov V.S. Arifmetiko-konveyernye vychisleniya v dvumernykh odnorodnykh sredakh // Materialy 3 Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii «Novye informatsionnye tekhnologii i sistemy» 10-11 dekabrya 1998 g., g. Penza, – Penza, 1998, S. 53-54.
6. Knyaz’kov V.S. Obshchaya otsenka slozhnosti realizatsii massivno-kletochnykh vychisleniy v iteratsionno-bitovykh vychislitel’nykh strukturakh s mnogomernoy organizatsiey. //Izvestiya TRTU, N 3,1997,c.218.
7. Knyaz’kov V.S. Dvumernye iterativno-bitovye protsessory: vremennaya i prostranstvennaya slozhnost’ posledovatel’no-massivnykh iteratsionno-razryadnykh vy-

chisleniy //Materialy 2 -oy Mezhdunar. n/t konf. «Novye informatsionnye tekhnologii i sistemy,ch.1,1996, Penza,S.108-109.

8. Knyaz'kov V.S., Biktashev R.A. Arkhitektura parallel'nykh vychislitel'nykh sistem.-Penza,1993.-s.166.
9. Knyaz'kov V, S., Volchenskaya T. V. Sposoby postroeniya konveyernykh vychislitel'nykh struktur s upravleniem kommutatsii potokov dannykh. Dep. ruk., VINITI, N 5581V9D ot 31. 10. 90, Minvuz RSFSR, Penza, Penzenskiy politekhnicheskiy i – nt, 1990, 10 s.