

Оптимизация параллельных вычислений бортовых систем реального времени

Часть 2

С. Назаров, д. т. н.¹, А. Барсуков, к. т. н.²

УДК 681.3.012 | ВАК 05.13.01

Во второй части статьи приводится пример оптимизации параллельного вычислительного процесса в бортовых вычислительных системах. Показано, что эффективное планирование параллельного вычислительного процесса в БВС можно обеспечить, используя методы сетевого планирования и управления в совокупности с математическим аппаратом ясно-параллельных графов и математического программирования.

РЕШЕНИЕ ЗАДАЧИ

Минимизация количества ресурсов выполнения пакета ЗНЗ без увеличения длины критического пути

Будем считать, определенный выше критический путь, равный 150 единицам времени, соответствует директивному времени решения заданного пакета задач. В этом случае становится актуальной задача минимизации ресурсов (в нашем случае количества процессоров), необходимых для реализации всех задач при условии непревышения директивного значения длины критического пути. Другими словами, длительности выполнения всех работ пакета нужно изменить так, чтобы длина любого пути в графе была в идеале равна или меньше длины критического пути. Практически все вычислительные работы будут увеличены с учетом возможных резервов времени для их выполнения.

Для формализации задачи введем дополнительные обозначения:

- t_{ij}^x – время выполнения работы (i, j) после минимизации количества выделяемых ресурсов;
- T_i^x – время свершения событий в графе работ (события отождествляются с вершинами графа);
- V_{ij} – исходное число процессоров, которое позволяет выполнить все работы с учетом возможного параллельного выполнения задачи;
- N_m – минимальное количество процессоров, которое будет получено в результате решения оптимизационной задачи (пока еще без учета возможности параллельной работы различных задач ЗНЗ);
- d_{ij} – доля физического вычислителя соответствующего виртуального процессора, необходимая для

выполнения работы (i, j) после минимизации количества выделяемых ресурсов,

$$d_{ij} = V_{ij} - V_{ij} \frac{(t_{ij}^x - t_{ij})}{t_{ij}^x}.$$

В качестве исходного количества вычислителей можно принять сумму V_{\min} (вычисляется по формуле (2) из первой части статьи $V_{\min} \geq T_3 / T_{kr} = 4$) с дополнительным числом вычислителей, которые надо добавить для выполнения параллельных ветвей задач ЗНЗ (см. табл. 3, первая часть статьи). Таким образом, $N = 4 + 11 = 15$. Значение минимального количества процессоров определяется как сумма долей физических процессоров в виртуальных вычислителях, то есть

$$N_{\min} = \sum_{i=0}^{i=6} \sum_{j=1}^{j=7} d_{ij}.$$

В качестве целевой функции задачи выбираем время занятости процессоров (полное машинное время) на решение пакета задач ЗНЗ. Его нужно минимизировать за счет использования имеющихся резервов времени на выполнение отдельных задач. Таким образом, необходимо найти такие значения множества переменных t_{ij}^x выполнения работ (i, j) , которые минимизируют целевую функцию времени выполнения ЗНЗ

$$T_{\text{ЗНЗ}} = \sum_{i=0}^{i=6} \sum_{j=1}^{j=7} t_{ij}^x - \sum_{i=0}^{i=6} \sum_{j=1}^{j=7} (t_{ij}^x - t_{ij}) \rightarrow \min. \quad (12)$$

Первое слагаемое этой функции представляет собой полные затраты машинного времени на решение всего пакета задач. Второе – экономию машинного времени при условии того, что можно увеличить время решения некоторых задач за счет имеющегося резерва времени на их выполнение (здесь $(t_{ij}^x - t_{ij}) \geq 0$). Рассмотрим ограничения, которые должны учитываться при решении этой задачи.

¹ ЗАО «МНИТИ», главный научный сотрудник, действительный член Международной академии информатизации.

² ЗАО «МНИТИ», заместитель генерального директора, профессор Академии военных наук, действительный член Международной академии безопасности.

Работа (i,j)	Продолжительность работы t_{ij}	Общий резерв времени работы R_{ij}	$t_{ij} + R_{ij}$	V_i^*	$V_i^* + t_{ij}$	Исходное число вычислителей V_i	Минимальное число вычислителей V_i^*	Путь	Длина пути
«0, 1»	30,00	0,00	30,00	30,00	0,00	1,00	1,00	0-2-7	150,00
«0, 2»	40,00	70,00	110,00	75,00	35,00	2,00	1,07	0-3-5-7	150,00
«0, 3»	50,00	25,00	75,00	75,00	25,00	1,00	0,67	0-1-6-7	150,00
«1, 3»	30,00	15,00	45,00	45,00	15,00	1,00	0,67	0-1-3-5-7	150,00
«1, 4»	35,00	0,00	35,00	35,00	0,00	2,00	2,00	0-1-4-5-7	150,00
«1, 6»	45,00	35,00	80,00	80,00	35,00	3,00	1,69	0-1-4-6-7	150,00
«2, 7»	40,00	70,00	110,00	75,00	35,00	3,00	1,60		
«3, 5»	40,00	35,00	55,00	40,00	0,00	3,00	3,00		
«4, 5»	40,00	10,00	50,00	50,00	10,00	2,00	1,60		
«4, 6»	45,00	10,00	55,00	45,00	0,00	2,00	2,00		
«5, 7»	35,00	0,00	35,00	35,00	0,00	2,00	2,00		
«6, 7»	40,00	0,00	40,00	40,00	0,00	1,00	1,00		

Т_{ит} = 315,00

Рис. 5. Решение задачи минимизации вычислительных ресурсов

Первый вид ограничений связан с принятым условием использования для решения задачи только имеющихся резервов для выполнения задач пакета. Отсюда следует система ограничений для продолжительности выполнения работ следующего вида:

$$t_{ij} + R_{ij} \geq t_{ij}^* \geq t_{ij} \quad (13)$$

Второй вид ограничений должен обеспечить такое изменение длительности выполнения всех работ пакета, чтобы длина любого пути в графе была в идеале равна длине критического пути. В рассматриваемом примере таких путей шесть (перечислим их последовательно вершинами графа): P1: 0–2–7; P2: 0–3–5–7; P3: 0–1–6–7; P4: 0–1–3–5–7; P5: 0–1–4–5–7; P6: 0–1–4–6–7. Ограничения на длину этих путей имеют следующий вид:

$$L_i(P_i) \leq T_{kr} | i = 1, 2, \dots, M-1. \quad (14)$$

Решение задачи (12) – (14) показано на рис. 5.

Как видно из результата решения задачи минимизации использования ресурсов, число процессоров при назначении отдельного вычислителя на одну и только одну задачу для реализации ЗНЗ без увеличения длины критического составляет $N_m = 23$ физических процессоров (сумма по столбцу G на рис. 5). При этом физические вычислители могут простаивать значительное количество времени. Переход на виртуальные

процессоры более низкой производительности, но работающие без простоев, позволяет сократить число вычислителей до $N_{min} = 18,29$ виртуальных процессоров (сумма по столбцу H на рис. 5). Однако структура пакета (см. рис. 1) свидетельствует о возможности параллельной и мультипрограммной работы этих процессоров при выполнении задач пакета ЗНЗ.

Возможности организации мультипроцессорного выполнения пакета задач, представленного сетевой моделью

Потенциальную параллельность выполнения заданного набора задач можно определить, преобразовав граф задач в ярусно-параллельную форму [7]. Ярусно-параллельная форма графа (ЯПФ) – деление вершин ориентированного ациклического графа на перенумерованные подмножества V_j , такие, что, если дуга идет от вершины $v_i \in V_j$ к вершине $v_m \in V_k$, то обязательно $j < k$.

Каждое из множеств V_j называется ярусом ЯПФ, j – его номером, количество вершин $|V_j|$ в ярусе – его шириной. Количество ярусов в ЯПФ называется ее высотой, а максимальная ширина ее ярусов – шириной ЯПФ. Для ЯПФ графа важным является тот факт, что операции, которым соответствуют вершины одного яруса, не зависят друг от друга (не находятся в отношении связи), и поэтому заведомо существует параллельная реализация алгоритма, в которой они могут быть выполнены параллельно на разных устройствах вычислительной системы. Поэтому ЯПФ графа алгоритма может быть использована для подготовки такой параллельной реализации алгоритма.

Для получения ЯПФ пакета задач, представленного в форме сетевой модели, как показано на рис. 1, его предварительно необходимо преобразовать, заменив дуги графа

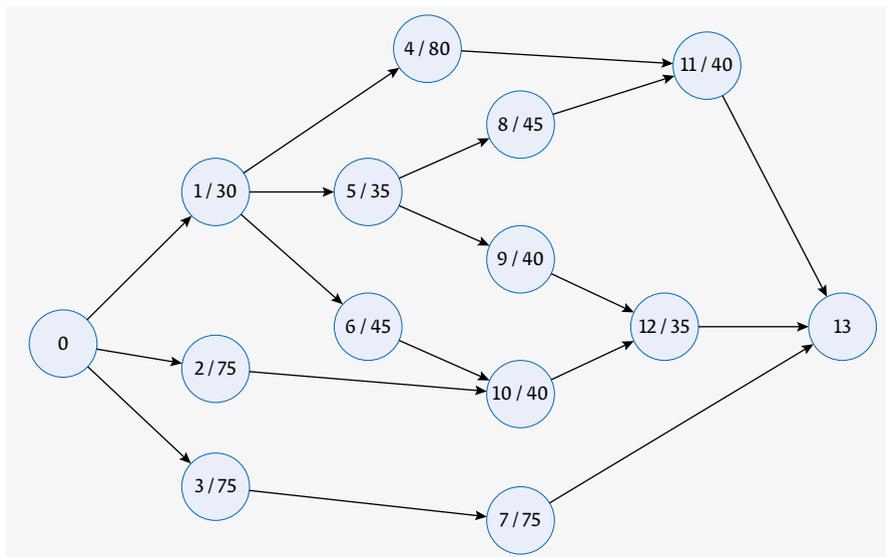


Рис. 6. Преобразованный граф пакета задач приложения

(работы) вершинами. В преобразованном графе (рис. 6) номера вершин соответствуют работам, длительность которых пересчитана в соответствии с решением задачи (12–14) по рис. 5.

Получить ЯПФ можно, построив матрицу смежности графа. Матрица смежности – это квадратная матрица размерностью $(M+1)(M+1)$, где M – число вершин графа, однозначно представляющая его структуру. Обозначим ее как $A = \|a_{ij}\|$, где каждый элемент матрицы определяется следующим образом: $a_{ij} = 1$, если есть дуга (i, j) , $a_{ij} = 0$, если нет дуги (i, j) . В нашем примере матрица смежности будет иметь следующий вид, приведенный на рис. 7.

Алгоритм распределения модулей системы по уровням:

1. Находим в матрице нулевые строки. В нашем случае это только одна строка с номером 13.
2. Вершина с этим номером образует нулевой (низший) уровень ЯПФ.
3. Вычеркиваем столбцы с номерами найденных вершин. В нашем случае – столбец 13.

		Номер вершины													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
Номер вершины	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	5	0	0	0	0	0	0	0	0	1	1	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	8	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	9	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	10	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	12	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рис. 7. Матрица смежности

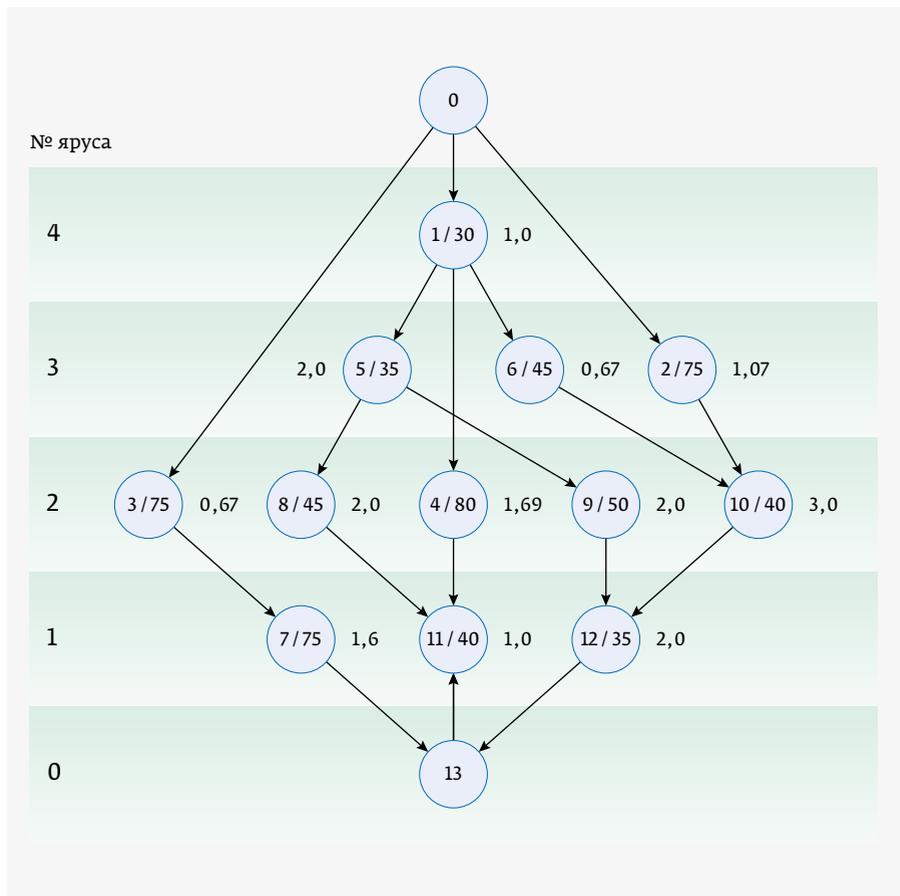
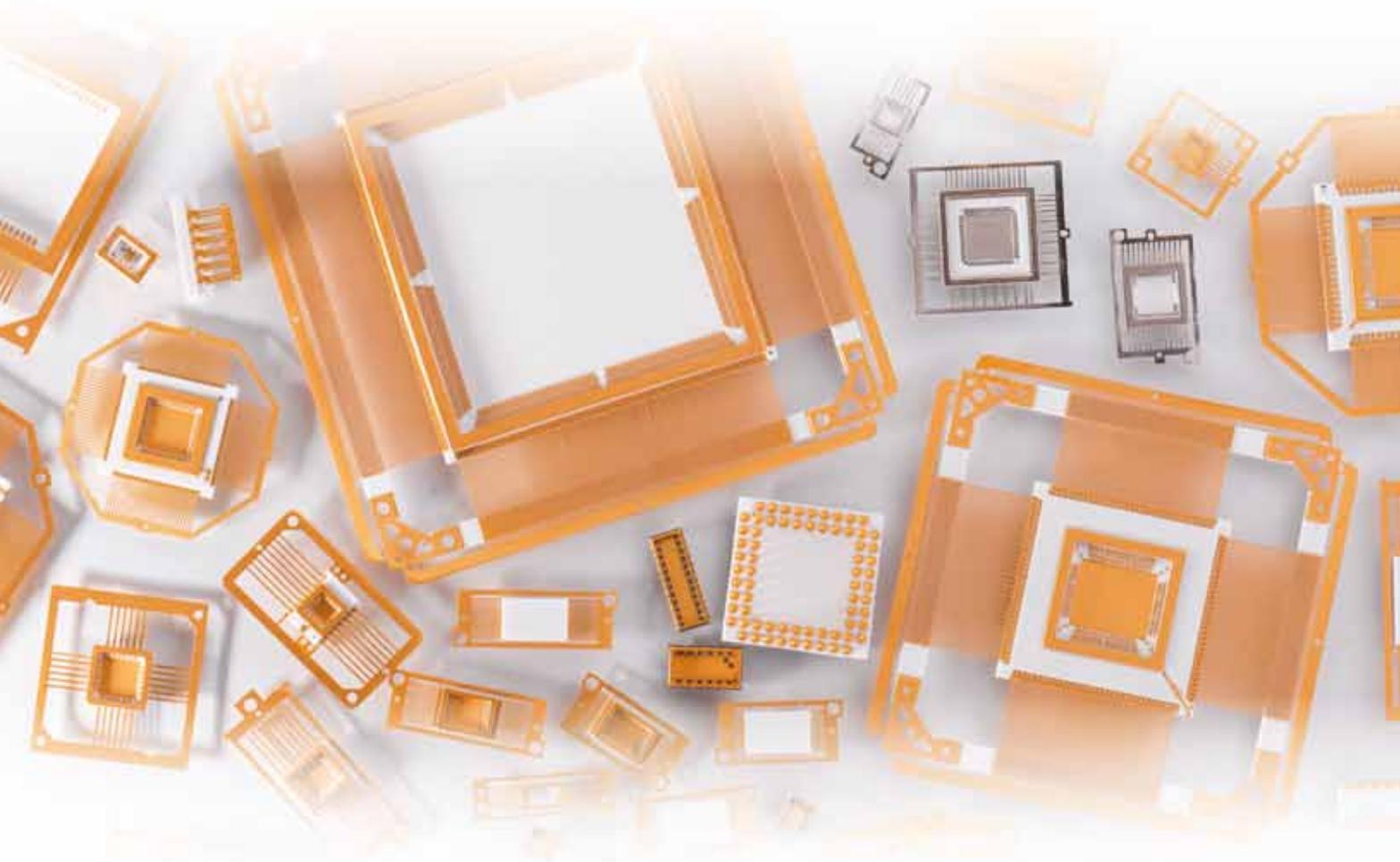


Рис. 8. ЯПФ графа пакета задач приложения

4. Находим в матрице нулевые строки (7, 11, 12). Это вершины 1-го уровня.
5. Вычеркиваем столбцы с номерами 7, 11, 12.
6. Находим в матрице нулевые строки (3, 4, 8, 9 и 10). Это вершины 2-го уровня.
7. Вычеркиваем столбцы с номерами найденных вершин.
8. Находим в матрице нулевые строки (2, 5, 6). Это вершины 3-го уровня.
9. Вычеркиваем столбцы с номерами 5, 6.
10. Вершина с номером 1 образует 4-й уровень.

ЯПФ графа задач пакета, полученная на основе матрицы смежности, представлена на рис. 8. В каждой вершине графа в виде дроби указан номер вершины (числитель) и время выполнения (знаменатель). Рядом с вершиной указано количество виртуальных процессоров, которые необходимы для выполнения задачи соответствующего номера.

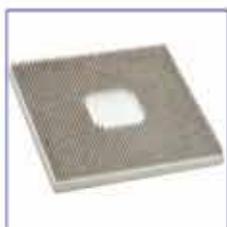
Построенный граф задач в ЯПФ далеко не прямоугольный, что неудобно для организации параллельного вычислительного процесса и определения минимально необходимого количества вычислителей. Визуально из рис. 8 понятно, что граф можно легко перестроить, не меняя связей между вершинами, например, можно



Предприятие располагает научно-технической и испытательной базой для проведения исследований, разработки и выпуска новой продукции



Выводные рамки



Металлокерамические корпуса



Нагревательные элементы



Контактные устройства



Графитовая оснастка



Оптоэлектронные корпуса



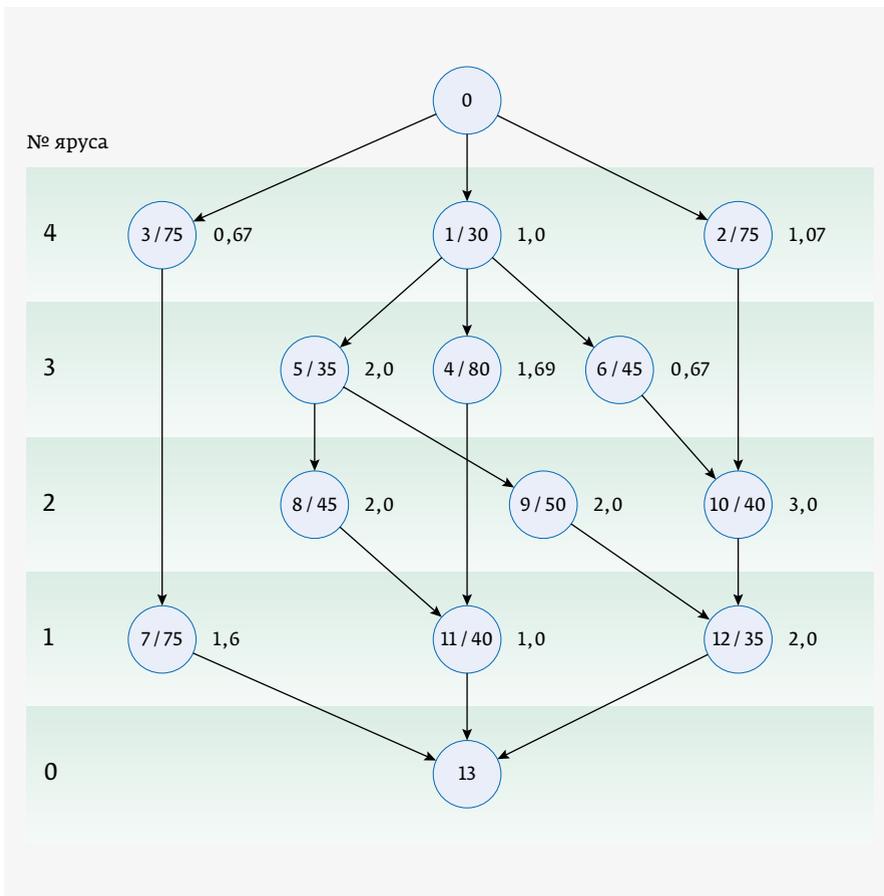


Рис. 9. Преобразованный ЯПФ пакета ЗНЗ

переместить вершины 2 и 3 на 4-й уровень, а вершину 4 – на 3-й. После таких преобразований граф примет вид, показанный на рис. 9.

По ЯПФ (рис. 9) легко построить план реализации вычислительного процесса во времени. На рис. 10 сверху показана шкала времени (равная длине критического пути), под которой обозначены временные промежутки реализации задач пакета. В нижней части диаграммы приведены сведения о необходимом количестве виртуальных вычислителей. Красными стрелками показаны передачи данных между задачами ЗНЗ. В нашем примере минимальное количество виртуальных вычислителей $V_{min} = 9,6$. В этом случае можно получить минимально возможное время решения системой ЗНЗ со значением $T_{ЗНЗ}^{min} = 150$ условных единиц. Для этого результата БВС должна содержать $N_m = 10$ физических процессоров.

Минимизация загрузки БВС

Как показывает диаграмма по рис. 10, требуемое количество вычислителей для реализации ЗНЗ с требуемым минимальным значением $T_{ЗНЗ} = 150$ условных единиц процессорного времени равно 10. При этом полная величина затрат процессорного времени БВС составит 1500 условных единиц. Реальные затраты (без учета проста процессоров) легко определить по рис. 10. Здесь можно выделить четыре этапа выполнения задач

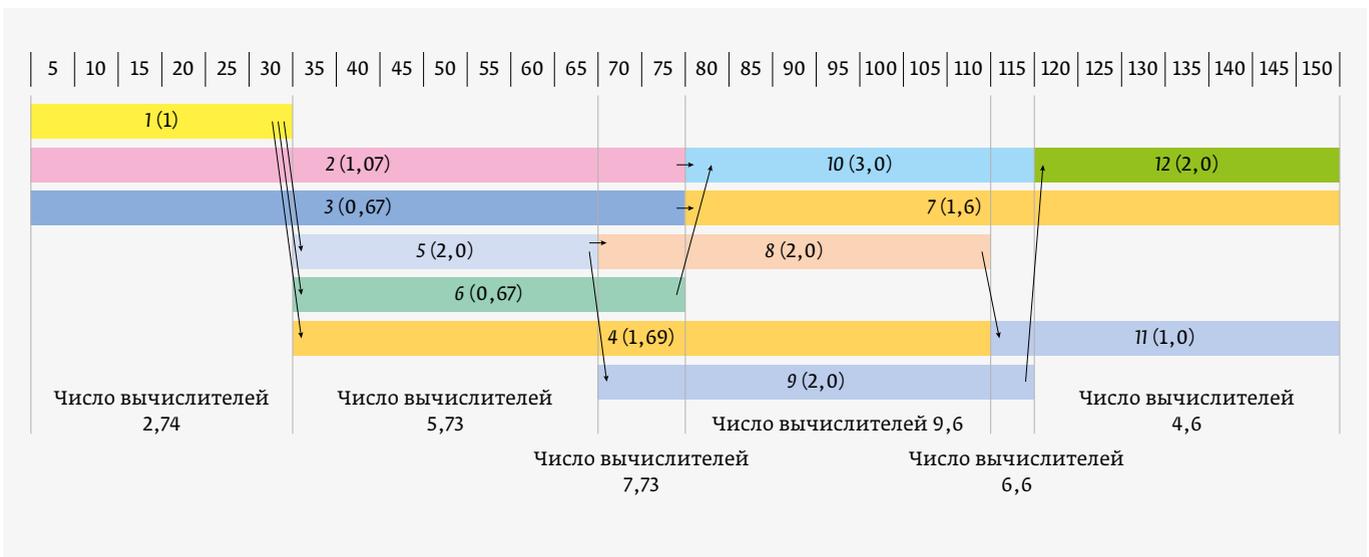


Рис. 10. Диаграмма выполнения вычислительных работ пакета ЗНЗ

МОНОЛИТ

www.monolit.by

ВИТЕБСКИЙ ЗАВОД РАДИОДЕТАЛЕЙ

**МНОГОСЛОЙНЫЕ
КЕРАМИЧЕСКИЕ
КОНДЕНСАТОРЫ**

ИМПОРТОЗАМЕЩАЮЩАЯ ПРОДУКЦИЯ

для высоконадёжной аппаратуры

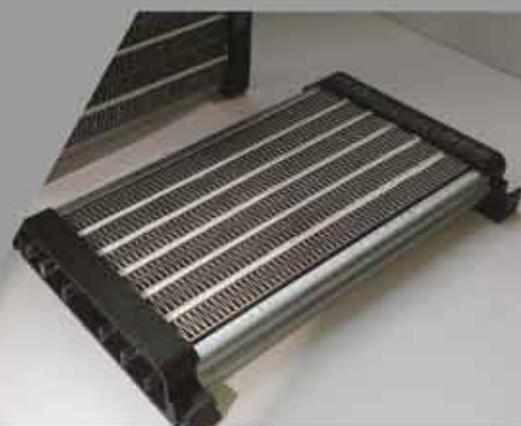
ТЕРМОРЕЗИСТОРЫ

с положительным температурным
коэффициентом сопротивления

РЕГИСТРЫ НАГРЕВАТЕЛЬНЫЕ

210101, РЕСПУБЛИКА БЕЛАРУСЬ
г. Витебск, ул. М. Горького, 145

Отдел маркетинга и сбыта:
тел.: + 375 (212) 36-44-52;
+ 375 (212) 36-45-34
факс: + 375 (212) 36-44-65
e-mail: monolmarket@mail.ru
monosbet@mail.ru



Акционерное общество
«СПЕЦ-ЭЛЕКТРОНКОМПЛЕКТ»

www.monolit.by

**ЭКСКЛЮЗИВНЫЙ ДИЛЕР
НА ТЕРРИТОРИИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Почтовый адрес: 125319, г. Москва, а/я 92.
Офис: г. Москва, ул. Краснопролетарская, д. 16, стр. 2
+7 (495) 234-01-10, факс: +7 (495) 956-3346
sales@zolshar.ru

Таблица 5. Затраты процессорного времени на выполнение ЗНЗ

	№ этапа						Суммарное время
	1	2	3	4	5	6	
Продолжительность этапа	30	35	10	35	5	35	
Количество вычислителей	2,74	5,73	7,73	9,6	6,6	4,6	
Затраты процессорного времени	82,2	200,55	77,3	336	33	161	890,05

пакета ЗНЗ. Каждый этап характеризуется неизменным количеством вычислителей и продолжительностью. Так, на первом этапе выполнения ЗНЗ занято 2,74 вычислителя, а продолжительность этапа – 30 условных единиц. Таким образом, затраты этого этапа составляют $30 \cdot 2,74 = 82,0$ единиц процессорного времени. Аналогично вычисляются затраты процессорного времени на последующих этапах (табл. 5).

Полагая, что ПЛИС вычислителей обладает свойством реконфигурации, имеет смысл заранее запрограммировать переключения освободившихся от выполнения текущей задачи процессоров на последующие задачи согласно диаграмме выполнения работ ЗНЗ. Другими словами, надо составить расписание загрузки и переключения процессоров. Постановка и формализация задачи построения такого расписания достаточно сложна и могла бы быть темой отдельной статьи. Однако в нашем небольшом примере можно легко увидеть возможности уменьшения простоя вычислителей. После завершения задачи 1 (желтый цвет на диаграмме) освободившийся вычислитель можно назначить на выполнение задачи 5, затем 9 и 12. Таким образом, этот вычислитель будет работать без простоя.

* * *

Приведенные в статье математические модели и пример оптимизации параллельного вычислительного процесса в бортовых вычислительных системах позволяет сделать следующие выводы:

1. Эффективное планирование параллельного вычислительного процесса в бортовых вычислительных системах можно обеспечить, используя методы сетевого планирования и управления в совокупности с математическим аппаратом ярусно-параллельных графов и математического программирования.
2. Реализация в программируемых логических интегральных схемах ПЛИС процессорных элементов

с разделяемой производительностью позволяет удобно использовать аппарат виртуальных процессоров для решения задачи распараллеливания вычислительного процесса и минимизации физических ресурсов для выполнения ЗНЗ с заданными требованиями по времени выполнения.

3. Возможности программируемой реконфигурации разрабатываемых в настоящее время ПЛИС процессорных элементов с разделяемой производительностью целесообразно использовать для построения эффективных расписаний загрузки процессоров БВС.

ЛИТЕРАТУРА

1. **Гохрингер Д., Хюбнер М., Бекер Ю.** Архитектура адаптивных многопроцессорных систем на кристалле: новая степень свободы при проектировании систем и в поддержке при выполнении // Мир радиоэлектроники. М.: ТЕХНОСФЕРА, 2012. С. 146–173.
2. **Хитт Д.** Стратегия Xilinx – быстрее двигаться к адаптируемому, интеллектуальному миру // ЭЛЕКТРОНИКА: Наука, Технология, Бизнес. 2018. № 4. С. 84–87.
3. Xilinx Ultrascale + Обзор. [Электронный ресурс]. URL: https://developer.ridgerun.com/wiki/index.php?title=Xilinx_Ultrascale%2B_Overview
4. **Кофман А. В.** Сетевые методы планирования: Применение системы ПЕРТ и ее разновидностей при управлении производственными и научно-исследовательскими проектами / Пер. с фр. М.: Прогресс, 1968. 181 с.
5. Сетевой график. [Электронный ресурс]. URL: <http://www.stroitelstvovone.ru/1/sete-voy-grafik.shtml>.
6. **Вagner Г.** Основы исследования операций. Том 1 / Пер. с англ. М.: МИР, 1972. 336 с.
7. **Назаров С. В.** Операционные системы специализированных вычислительных комплексов: теория построения и системного проектирования. М.: Машиностроение, 1989. 400 с.



Разработка и производство конденсаторов

оксидно-электролитические алюминиевые конденсаторы
K50-15, K50-17, K50-27, K50-37, K50-68, K50-77, K50-80, K50-81, K50-83,
K50-84, K50-85, K50-86, K50-87, K50-88, K50-89, K50-90, K50-91, K50-92,
K50-93, K50-94, K50-95(чип), K50-96, K50-97(чип), K50-98, K50-99,
K50-100, K50-101, K50-102, K50-103

объемно-пористые танталовые конденсаторы
K52-1, K52-1М, K52-1БМ, K52-1Б, K52-9, K52-11,
K52-17, K52-18, K52-19, K52-20, K52-21, K52-24,
K52-26(чип), K52-27(чип), K52-28

оксидно-полупроводниковые танталовые конденсаторы
K53-1А, K53-7, K53-65(чип), K53-66, K53-68(чип),
K53-69(чип), K53-71(чип), K53-72(чип),
K53-74(чип), K53-77(чип), K53-78(чип), K53-82

суперконденсаторы (ионисторы)
K58-26, K58-27, K58-28, K58-29, K58-33

накопители электрической энергии на основе модульной сборки суперконденсаторов МИК, МИЧ

Система менеджмента качества сертифицирована на соответствие требованиям ISO 9001

Россия, 427968, Удмуртская Республика, г. Сарапул, ул. Калинина, 3
Тел.: (34147) 2-99-53, 2-99-89, 2-99-77, факс: (34147) 4-32-48, 4-27-53
e-mail: elecond-market@elcudm.ru, <http://www.elecond.ru>