УДК 519.6

Кобайло А. С., доцент

ЭЛЕМЕНТЫ ТЕОРИИ СИНТЕЗА ВЫЧИСЛИТЕЛЬНЫХ СТРУКТУР РЕАЛЬНОГО ВРЕМЕНИ ДЛЯ ПОЛИГРАФИЧЕСКИХ КОМПЛЕКСОВ

The paper presents a discussion of implementation of on-line computational structures synthesis theory in the field of printing complex. Theory principal terms are determination of time hierarchy levels through peaks of computational algorithm graph (CAG) of a printing complex and feasibility of CAG route in on-line version

Введение. Современный полиграфический комплекс представляет собой сложную аппаратно-программную систему, которую можно представить в виде множества взаимодействующих структурных компонентов – функциональных устройств (ФУ). Проектирование таких комплексов предполагает построение избыточной структуры системы или синтез всех возможных ее вариантов и последующий выбор оптимальных по заданным критериям качества. Многообразие альтетнативных вариантов и недостатки эвристических методов проектирования обусловливают необходимость применения формальных методологий синтеза систем, ориентированных на возможность автоматизации проектирования. В качестве такой методологии может быть использована теория синтеза вычислительных структур реального времени.

Основная часть. Теория синтеза вычислительных структур реального времени базируется на двух основных положениях: назначении уровней временной иерархии вершинам графа вычислительного алгоритма (ГВА) реализации математической модели проектируемой системы и условии реализуемости пути ВГА в реальном времени [2]. Методология, основанная на данной теории, предполагает реалиизацию последовательности процедур синтеза вычислительных структур реального времени.

Условно среди ряда процедур синтеза можно выделить четыре группы соответственно четырем условным этапам синтеза:

исследования графа базовой структуры (ГБС);

разработка вычислительного графа алгоритма (ВГА);

формирование вектора временной развертки вычислительного графа алгоритма;

построение графа вычислительной структуры (ГВС).

Рассмотрим основные положения, на которых основаны процедуры двух первых этапов.

1. Основной целью исследования графа базовой структуры является определение ФУ, которые должны реализовывать операции, отождествленные с соответствующими вершинами графа вычислительного алгоритма.

ОПРЕДЕЛЕНИЕ 1.1. Полным путём L графа алгоритма называется путь, связывающий одну из начальных вершин графа с одной из его конечных вершин.

ОПРЕДЕЛЕНИЕ 1.2. Прямоугольная матрица D размером $n \times s$, где n – множество вершин графа базовой структуры, s – множество ФУ из заданного набора, для которой

$d_{ij} = \begin{cases} 1, \\ 0 \end{cases}$ если j-е ФУ реализует операцию, отождествленную с i-й вершиной графа. Во всех остальных случаях называется матрицей соответствия.

УТВЕРЖДЕНИЕ 1.1. Если вершина v_i принадлежит множеству вершин $V(\gamma) \leftrightarrow \Delta t(\gamma)$, то данной вершине может быть назначено ФУ, для которого выполняется условие

$$\tau_i^j \le \Delta t(\gamma), \qquad (1)$$

где τ_i^j – время обработки j-м ФУ операции, отождествлённой с вершиной v_i

УТВЕРЖДЕНИЕ 1.2.

Если $S_i = \left\{ \varsigma_j^{(i)}, i = \overline{1, I}, j = \overline{1, J_i} \right\}$ – множество ФУ, реализующих операцию $\varphi_i \leftrightarrow v(i) \leftrightarrow \Delta t(\gamma)$ последовательной обработки потока данных и $\forall i, j : \tau_i^j \succ \Delta t(\gamma)$, то последовательность вершин $\{v(.)\}$ может быть отображена на ПКВ с длительностью цикла $\tau_K = \Delta t(\gamma)$ и количеством ступеней $\rho = \left[\tau_i^j / \Delta t(\gamma) \right]$ для выбранного ФУ.

УТВЕРЖДЕНИЕ 1.3. В случае отображения последовательности вершин ГБС на ПКВ каждой из этих вершин может быть назначено любое из ФУ, способных выполнить соответствующую операцию за заданное время.

Отметим, что каждому вектору назначения соответствует свой вектор реализации, элементы которого $\{\tau_i\}$ формируются как параметры τ_i^j соответствующих ФУ.

2. Этап разработки вычислительного графа алгоритма базируется на введении буферной памяти, определении конвейеризируемых путей, свертываемых вершин, условий добавления вершин.

УТВЕРЖДЕНИЕ 2.1. Если для дуги (v_i, v_j) графа вычислительного алгоритма справедливо $v_i \in V(\gamma_i), v_j \in V(\gamma_j), \gamma_i \succ \gamma_j$, то для данной дуги должна быть выполнена операция добавления вершины, которая отождествляется с операцией хранения данных операции хранения данных на интервале $\Delta t(\gamma_i)$.

УТВЕРЖДЕНИЕ 2.2. Вершине, отождествленной с операцией хранения данных и инцидентной двум вершинам v_i и v_j с разным уровнем временной иерархии, должен быть назначен уровень временной иерархии, соответствующий наивысшему из уровней временной иерархии вершин v_i и v_j.

ОПРЕДЕЛЕНИЕ 2.1. Графом алгоритма с буферной памятью (ГАБП) называется граф, сформированный из ГВА путём последовательного выполнения операции добавления вершины для дуг, связывающих вершины с разными уровнями временной иерархии.

Каждой из данных вершин назначается блок памяти из условия $\tau_g^j \le \Delta t(\gamma)$, где τ_g^j – время доступа (цикл запись-чтение) j-го ФУ памяти, $\Delta t(\gamma) = \min_{i,j}\{\Delta t(\gamma_i), \Delta t(\gamma_j)\}$.

Обозначим, в отличие от ГВА, ГАБП как G^{bs}. Для этого графа формируется спецификация путем расширения спецификации ГВА; каждая новая вершина отождествляется с операцией хранения. Строятся также новый вектор назначения $\vec{R}^{bs}$ и новый вектор реализации $\vec{\tau}^{bs}$ путём введения соответствующих элементов в вектора $\vec{R}$ и $\vec{\tau}$ ГВА.

ОПРЕДЕЛЕНИЕ 2.2. Базисным путём называется путь L(1), максимальный по времени реализации из множества полных путей первого уровня временной иерархии при отсутствии циклов у этих путей, или наибольший из циклов первого уровня временной иерархии $L_c(1)$:

$$L_B = \{L^n(1), L_c(1)\} : L_B \leftrightarrow T(L_B),$$

$$T(L_B) = \begin{cases} \max\limits_k \{T_k^v(1)\}, k = \overline{1, K_1} \\ \max\limits_n \{T_n^c(1)\}, n = \overline{1, N_1}, \end{cases}$$

$$V_1(1) \cap V_2(1) \cap \ldots \cap V_{k1} = V(1),$$

где $T_k^v(1)$ – время реализации k-го пути первого уровня временной иерархии, $T_n^c(1)$ – время реализации наибольшего из циклов для путей первого уровня временной иерархии, $V_k(1) | k = \overline{1, K_1}$ – множество вершин k-го пути первого уровня временной иерархии, K_1 – количество полных путей первого уровня временной иерархии, N_1 – количество циклов для путей первого уровня временной иерархии.

ОПРЕДЕЛЕНИЕ 2.3. J-м усеченным путём γ-го уровня временной иерархии называется путь γ-го уровня временной иерархии, для множества вершин V которого справедливо

$$V(j,\gamma) = V_L(\gamma) \setminus \bigcup_{k=1}^{j-1} V(k,\gamma)$$

ПРИМЕЧАНИЕ 2.1. В дальнейшем вместо указанного термина возможно использование термина "усеченный путь". Базисный путь переносится во множество усеченных путей без преобразований.

УТВЕРЖДЕНИЕ 2.3. Для цикла конвейера τ_K справедливо

$$\max_j \{\tau^{(j)}\} \le \tau_K \le \Delta t(\gamma),$$

где $\tau_K^{(j)}$ – время выполнения операции j-м ФУ, включенным в КВ, $\Delta t(\gamma)$ – шаг дискретизации для вершин конвейеризируемого пути.

Отметим, что для синхронной работы конвейера целесообразно выбирать $\tau_K = \Delta t(\gamma)$.

УТВЕРЖДЕНИЕ 2.4. В ступень конвейера должны входить ФУ, для которых суммарное время выполнения операций меньше или равно его циклу.

УТВЕРЖДЕНИЕ 2.5. Усеченный путь графа алгоритма L_γ, для которого не выполняется условие его реализуемости в реальном времени, может быть отображён на конвейерный вычислитель, если

ОПРЕДЕЛЕНИЕ 2.4. Условно-конвейеризируемым путём γ-го уровня временной иерархии L будем называть совокупность вершин $\{V_{(i)}\}, i = \overline{1, I}$, относящихся ко множеству неконвейеризируемых путей γ-го уровня временной иерархии, расположенных в порядке возрастания значения координаты вектора временной развертки t(i). Обозначим I_γ – мощность множества вершин неконвейеризируемых путей уровня γ.

ОПРЕДЕЛЕНИЕ 2.5. Условной μ-й ступенью конвейера называется совокупность вершин условно-конвейеризируемого пути L_r^y, для которой справедливо

$$\min_{l\le j\le l_r}\{t(j)\}=t(1)+(\mu-1)\times\tau_K,$$

$$\max_{l\le j\le l_r}\{t(j)+\tau(j)\}-\min_{l\le j\le l_r}\{t(j)\}\le t(\gamma),$$

$$e: v(1),v(2),...,v(l-1)\subset\bigcup_{i=1}^{\mu-1}V^{(i)},$$

где v(1),…,v(I-1) – последовательные вершины пути L_r^y, $V^{(i)}$ – множество вершин i-й условной ступени. Условно-конвейеризируемые пути вводятся для упрощения и регулярности процедур построения блоков управления вычислительной структурой (см. раздел 3).

ОПРЕДЕЛЕНИЕ 2.6. Множеством свертываемых вершин S_i называется множество мощностью больше единицы вершин ГБП или вычислительного графа алгоритма, являющихся прообразами одной из вершин графа вычислительной структуры.

УТВЕРЖДЕНИЕ 2.6. Операция элементарного гомоморфизма может быть выполнена над вершинами u ,v графа G=(V,E), если этим вершинам назначены ФУ одного типа при выполнении условия $t_u+\tau_u\le t_v$, на одном цикле обработки данных вершинами u и v , где t_u и t_v – моменты начала выполнения операций, отождествленных с вершинами u и v соответственно, τ_u – время реализации вершины u.

Выполнение операции элементарного гомоморфизма подразумевает реализацию двух или более операций одним и тем же ФУ. Очевидно, что это возможно только в тех случаях, когда соответствующие операции реализуются на взаимно непересекающихся интервалах времени, из чего следует справедливость утверждения.

СЛЕДСТВИЕ 2.1. Свертываемым может быть любое количество вершин, которым назначены ФУ одного типа, принадлежащих одному усеченному пути и не являющихся вершинами разных ступеней конвейера.

Определение вычислительного графа алгоритма (ВГА) приведено в. [1]. При этом условиями добавления вершин, отождествленных со служебными операциями, при синтезе вычислительных устройств являются: необходимость преобразования уровней сигнала, если две смежные вершины реализованы на различной элементной базе с разными уровнями логических сигналов, преобразование типа сигналов (цифровой – аналоговый и наоборот) и т.д. Рассмотрим ещё один типичный случай, характерный как для вычислительных устройств, так и вычислительных систем.

УТВЕРЖДЕНИЕ 2.7. Операция добавления вершины должна выполняться для дуг, входных по отношению к вершинам, включенным в одно из множеств свертываемых вершин, если начальным вершинам этих дуг не назначены ФУ с магистральным выходом. Выполнение операции дополнения вершины в соответствии с последним утверждением подразумевает расширение множества множеств свертываемых вершин, в которое включаются множества, каждое из которых составляют вершины, введённые в дуги, инцидентные одноименным входам вершин, включенных в одно из множеств свертываемых вершин согласно утверждению 2.6.

УТВЕРЖДЕНИЕ 2.8. Операция добавления вершины должна быть выполнена для дуги, инцидентной вершине, которой назначен блок памяти, если этот блок не работает в режиме постоянного чтения.

Смежные вершины, отождествленные с операциями записи и чтения одного блока памяти, образуют множество свертываемых вершин.

Таким образом, на данном этапе с учётом утверждений 2.6, 2.7 и 2.8 происходит двойное преобразование графа вычислительного алгоритма:

$$G\xrightarrow{ДВ}G^{БП}\xrightarrow{ДВ}G^{ВГА},$$

где $G^{БП}$ – граф алгоритма с буферной памятью, ДВ – операция дополнения вершины, $G^{ВГА}$ – вычислительный граф алгоритма. При этом построение графа $G^{ВГА}$ сопровождается формированием спецификации его вершин, вектора назначения $\vec{R}$ и вектора реализации $\vec{\tau}^{ВГА}$.

УТВЕРЖДЕНИЕ 2.9. Необходимым условием реализуемости вычислительной структуры в реальном времени является выполнение для всех не-конвейеризируемых путей вычислительного графа алгоритма условия $\sum_{i\in L}\tau(i)\le\Delta t(\gamma)$, где L_y – усеченный путь уровня γ.

Рассмотренные положения обусловливают следующую последовательность процедур синтеза вычислительных структур реального времени на первых этапах проектирования

1) формирование графа вычислительного алгоритма (определение 1.1);

2) определение полных путей графа вычислительного алгоритма (определение 2.4);

3) назначение уровней временной иерархии вершинам графа вычислительного алгоритма (определения 2.1, 2.2, утверждение 2.1);

4) назначение ФУ вершинам графа базовой структуры (определения 2.7,2.8, утверждения 2.3-2.5);

5) формирование вектора реализации (определение 1.3);

6) формирование графа алгоритма с буферной памятью (определение 2.9, утверждения 2.6, 2.7);

7) формирование усеченных путей γ-х уровней временной иерархии (определения 2.10, 2.11);

8) определение конвейеризируемых и условно-конвейеризируемых путей и ступеней конвейеров (утверждения 2.8, 2.10, определения 2.12, 2.13);

9) определение множеств свертываемых вершин (определение 2.14, утверждение 2.11, следствие 2.2);

10) построение вычислительного графа алгоритма (определение 1.4, утверждения 2.12, 2.13);

11) первая проверка реализуемости вычислительной структуры реального времени (утверждение 2.14, следствие 2.3).

Заключение. Использование рассиотренных элементов тнории синтеза вычислительных структур реального времени при проектировании полиграфического оборудования позволит автоматизировать основные этапы проектирования за счет возможности автоматизации в силу высокой степени формализации данной методики, что обеспечит:

– снижение стоимости проектирования;
– сокращение сроков разработки и создания новых образцов полиграфической техники;
– возможность выбора из множества альтернативных вариантов проектируемых систем наиболее перспективных.

Сопряжение пакета прикладных программ, реализующих **все** (в т.ч. и рассмотренные в данной статье) процедуры синтеза с известными САПР позволит автоматизировать все стадии проектирования полиграфического оборудования.

Литература

1. Воеводин, В. В. Математические методы и модели в параллельных процессах /В. В. Воеводин — М.:,Наука. 1989.— 296 с.

2. Кобайло, А. С. Основы теории синтеза вычислительных структур реального времени/ А.С. Кобайло — Минск: БГУИР. 2001. — 201 с.

3. Жиляк, Н. А. Базовый алгоритм синтеза вычислительных структур реального времени / Н. А. Жиляк, А. С. Кобайло // Труды БГТУ. Сер. VI, Физ.-мат. Науки и информ2007. Вып. XV. — С. 147–150.