

**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА**

Факултет:

**Изчислителна техника и автоматизация**

Катедра:

**Компютърни науки и технологии**

**Димитър Стоянов Тянев**

доц. д-р инж.

**СИНТЕЗ НА АСИНХРОННИ  
МИКРОКОНВЕЙЕРНИ СИСТЕМИ  
С ОБЩА СТРУКТУРА**

АВТОРЕФЕРАТ

на дисертация, представена за присъждане  
на научната степен "Доктор на науките"

в професионално направление

**5.3. КОМУНИКАЦИОННА И КОМПЮТЪРНА  
ТЕХНИКА**

специалност

**КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ**

Варна

2011

**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА**

Факултет:

**Изчислителна техника и автоматизация**

Катедра:

**Компютърни науки и технологии**

**Димитър Стоянов Тянев**

доц. д-р инж.

**СИНТЕЗ НА АСИНХРОННИ  
МИКРОКОНВЕЙЕРНИ СИСТЕМИ  
С ОБЩА СТРУКТУРА**

АВТОРЕФЕРАТ

на дисертация, представена за присъждане  
на научната степен "Доктор на науките"

в професионално направление

**5.3. КОМУНИКАЦИОННА И КОМПЮТЪРНА  
ТЕХНИКА**

специалност

**КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ**

Варна

2011

Дисертационният труд съдържа 308 страници, включващ 224 фигури и 17 таблици, оформен в предговор, 5 глави, основни резултати и списък на цитираната литература от 167 заглавия.

Защитата на дисертационния труд ще се състои на 18.09.2012 год. от 15<sup>00</sup> часа в зала "Конферентна" (НУК) на открито заседание на научно жури, сформирано със заповед на Ректора на ТУ-Варна № 156 от 20.03.2012 г.

#### Рецензенти:

1. Професор доктор на техническите науки Тодор Атанасов Стоилов - БАН, ИИКТ;
2. Професор доктор Пламенка Иванова Боровска - ТУ-София;
3. Професор доктор Ангел Сотиров Сфрикаров - РУ "А. Кънчев".

#### Членове на научното жури:

4. Професор доктор на техническите науки Райчо Тодоров Иларионов - ТУ-Габрово;
5. Професор доктор Нина Василева Синягина - ЮЗУ-Благоевград;
6. Академик доктор на техническите науки Кирил Любенов Боянов - БАН, ИИКТ;
7. Професор доктор на техническите науки Николай Димитров Минчев - ТУ-Варна.

Материалите по защитата (дисертация, рецензии и становища) са на разположение на интересуващите се във ФД "Докторанти", стая 318-НУК.

© Димитър Стоянов Тянев

© Технически университет - Варна, 2011.

#### **Обща характеристика на дисертационния труд**

Актуалност. Съвременното състояние на компютърните науки се характеризира с интензивно и многопланово развитие. Отделните главни направления се припокриват и активно си взаимодействат. Алгоритмично-програмните проблеми, обектите на приложение и дори данновите структури влияят върху апаратната архитектура на компютърните системи и обратно. Каквито и да са обаче взаимните връзки между отделните аспекти в това научно и технологично направление развитието на компютърните системи продължава да е подчинено на една главна цел – постигане на все по-високо бързодействие и производителност. Тази цел е особено значима в апаратно научно-технологичното направление.

Развитието на съвременните хардуерни решения се прилагат върху задачи, които досега са традиционно решавани със софтуерни средства. В същото време развитието на хардуерните системи въз основа на прилаганите за производството им технологии започват да изпитват затруднения, които са свързани с основния метод за управление – синхронния.

Настоящата дисертация е посветена на формулирането и принципното теоретично решаване на проблемите, свързани с хардуерната реализация на изчислителен процес с обща алгоритмична структура на микрооперационно ниво, търсейки по този начин възможности за хардуерните системи, които са присъщи на софтуерните.

Цел на изследването. Изследването цели разработване в рамките на единен подход на методологичната основа за синтез и анализ на хардуерна реализация с неконвенционална организация на изчислителен процес, характеризиращ се с произволна алгоритмична структура. То е фокусирано върху разработването на нови методи за апаратна реализация на функционално пълна система от алгоритмични структури и е ориентирано преди всичко към асинхронна и конвейерна организация на изчислителния процес. Освен конвейерна организация, са използвани и други форми на паралелизъм, като естествен и суперскаларен.

Методи на изследването. В дисертацията са използвани методите на логическия синтез и анализ, на теорията на крайните автомати, на синтез и анализ на алгоритми, на синтез и анализ на микроархитектури, елементи на математическия анализ, софтуерно моделиране, числено и логическо експериментирание.

Практическа приложимост. Получените алгоритмични и схемни решения могат да бъдат използвани в реални хардуерни системи без преработка, което е доказвано чрез техни реализации върху програмируеми интегрални схеми. Техническите реализации, които ги използват, са патентно защитими.

Място на изследването. Изследванията по дисертационния труд в основната си част са извършени във Факултета по изчислителна техника и автоматизация на Техническият университет – Варна.

Апробация на изследването. Резултати от дисертационното изследване са публикувани:

#### 1. В списания:

- *Applied Technologies and Innovations, ISSN 1804-1191, Prague Development Center, Czech Republic.*
- *Електро-промишленост и приборостроене, София.*
- *Компютърни науки и технологии, ISSN 1312-3335, ТУ-Варна.*

#### 2. В университетски научни сборници:

- *Научові праці ДонНТУ, Донецк, Україна, Серія "Інформатика, кібернетика та обчислювальна техніка".*
- *Годишник на ТУ-Варна, Юбилеен сборник "45 години ТУ-Варна".*

#### 3. На международни конференции:

- *International Conference on Computer Systems and Technologies (CompSysTech).*
- *International Workshop on Network and GRID Infrastructures, Bulgarian Academy of Sciences, Sofia, Bulgaria.*
- *Fourth International Bulgarian-Greek Conference (CS'08), Kavala, Greece.*
- *Fifth International Scientific Conference 2009 (CS'09), Sofia, Bulgaria.*
- *XLV International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST'2010), Ohrid, Macedonia.*
- *International Conference TELECOM'07, St. St. Constantin & Elena, Varna, Bulgaria.*

#### 4. В научно изследователски проекти на Технически университет - Варна.

Структура на дисертационния труд. Структурните елементи на дисертационния труд са предговор, 5 глави, обобщение на резултатите и списък на цитираните литературни източници (167 на брой). Текстът му е изложен на 308 страници. В него се съдържат 224 фигури и 17 таблици.

### Глава 1.

#### Аспекти на дисертационния проблем

От самото си зараждане, вече повече от шест десетилетия, развитието на компютърната техника протича в едно главно направление: стремеж към все по-големи скорости и производителност. Това главно направление е всеобхватно, защото то насочва научните изследвания както в областта на теоретичните методи, така и в областта на техническите средства за реализация. Ето защо и нашето изследване е ориентирано в това направление. В настоящата глава е представен преглед на основните аспекти на дисертационния проблем след анализ на достъпни литературни източници.

##### 1.1. Синхронност и Булева логика

Високата производителност на компютърната техника се преследва чрез множество подходи, но в основата стоят два алтернативни метода

на управление: синхронен и асинхронен. Компютърната техника от самото си раждане е асинхронна. Разработката на логически елементи, на логически възли, на устройства и на цели поколения изчислителни машини почти до средата на 70-те години е била асинхронна. Тактовите генератори не са били елемент в структурата на тази техника. В същото време развитието на интегралните технологии за производство на градивни елементи както и изискванията на пазара налагат масовото използване на синхронната техника.

Основното достоинство на синхронните схеми, което ги прави предпочитани, е предвидимият момент на превключване. Още през 1947 година Alan Turing е обосновал необходимостта от формално тактуване на последователностните схеми. Тактуването прави синтеза на логическите функции по-лесен, тъй като осигурява минималното им описание. Тактуваните схеми работят в дискретно и безмерно време. Синхронността се изразява в това, че превключването на даден логически елемент е възможно точно в момента, в който се появява тактовия импулс. Внасянето на такова условие за превключване значително облекчава синтеза на логическите системи.

Синхронните логически схеми по начало се синтезират "със запас". Това се разбира в следния смисъл: за осигуряване на надеждно функциониране по отношение на истинната логическа стойност в крайна сметка трябва да се отчете най-неблагоприятния случай. При такава нагласа, цената за правилното функциониране е времеви запас, т.е. изкуственото снижаване на бързодействието, което се осигурява чрез снижаване на тактовата честота. Достатъчно дългото (със запас) изчакване на схемата да се превключи и да установи на изхода си правилната логическа стойност гарантира надеждното ѝ функциониране и в най-неблагоприятните условия. Въпреки, че като правило допустимото захранващо напрежение може да варира в диапазона  $\pm 10\%$  от номиналното и при отчитане на ограниченията на останалите експлоатационни параметри, времевите загуби в синхронните системи могат да достигнат от 50% до 100% спрямо номиналните разчети при благоприятни условия. С други думи изчисленията се провеждат бавно, но сигурно.

Всяко превключване в стойността на една логическа функция преди момента на поява на тактовия импулс се приема за неистинно или още като част от още незавършилия преходен процес. Важна е логическата стойност непосредствено преди появата на тактовия импулс. Тя се приема за истинна, тя се фиксира и се използва за следващите изчисления. Следователно тактуването е трябвало да се съобрази единствено с времето, което е необходимо на логическите елементи да се превключат в истинната стойност.

Синхронната логика не съдържа в себе си условие за край на изпълнената операция. Тя може да съществува единствено на фона своята времедиаграма. С други думи, основните градивни логическите схеми "И" и "ИЛИ" трябва да имат още един управляващ вход, който се



рови възли и устройства са последователностните схеми. Главно място в тези схеми заемат различни запомнящи елементи, т.е. троични тригери. В подкрепа на асинхронната логика има разработки, които представяват тризначен еквивалент на Мюлер С-елемента.

Разработена е троична цифрова аритметика, като и необходимата за нейната хардуерна реализация троична логика, в това число суматори, броячи и други функционални възли. Следва да се подчертае, че троичната логика няма нищо общо с троичната бройна система. Това означава, че принципът на дуалност на Клод Шенон, който е в сила между Булевата логика и двоичната бройна система, тук не е в сила. Основните характеристики, които определят практическата ценност на троичната бройна система и на тризначната логика според Н.П. Брусенцов, са следните:

- Съществува естествена форма за представяне на цели числа със знак, т.е. не са необходими изкуствено въведените машинни кодове прав, обратен и допълнителен;
- Знакът на числото се съдържа (се изразява) чрез най-старшата значеща цифра и не е необходим специален знаков разряд;
- Числата се сравняват лесно, без да е необходимо да се взема под внимание техният знак;
- В резултат на това изпълнението на машинна команда за условен преход е два пъти по-бързо в сравнение с двоичната команда;
- Скъсяването на числото е равносилно на правилното му закръгляне; което двоичната бройна система не осигурява;
- В троичния суматор преноси възникват в около 33% от случаите на едноразрядно събиране (и по-точно в 8 случая от 27 възможни), докато при двоично събиране в 50% (в 4 от общо 8);
- Операции умножение и деление имат същата степен на сложност като двоичните. Умножение с (-1) сменя знака на числото;
- Логическият сигнал с три нива е по-защитен и по-устойчив срещу смущения, което означава, че специалните методи за троично кодиране с излишък е по-просто.

### 1.2.2. NULL Convention Logic (NCL)

В средата на 90-те години е предложен още един подход за синтез на асинхронна логика, наречен NCL-логика (*NULL Convention Logic*). През 2005 година излезе от печат книгата *“Logically Determined Design”* на Karl M. Fant изцяло посветена на този подход. За премахване на необходимостта от синхронизация в този подход се предлага интеграция на данните и управлението в един логически поток. В този поток разделителят (*null*) играе ролята на синхронизираща функция. Информационните (логическите) сигнали се предават парафазно едновременно по два проводника (*“1-true”, “0-false”*). Стойността на един сигнал се приема за истинна само във времето, когато по паралелния проводник се появи разделителят *“null”*, в което и се изразява синхронизацията. За целите на

синхронизацията се прилагат ключови елементи с хистерезис, чиито изходи могат да бъдат в две състояния – *“null”* и *“data”*. Ключовият елемент преминава в състояние *“data”* само когато всички входни стойности са *“data”*, съпроводени от своите разделители *“null”*.

Съвременен тласък в развитието на методите за синтез на схеми с използване на тази логика представя книгата на Scott Shmith и Jia Di *“Designing Asynchronous Circuits using NULL Convention Logic”* (2009 год.).

Според изследователите в тази група проблемите на Булевата логика могат да обобщат така:

- Логическите изрази не прилагат функционална зависимост, която да координира превключванията на логическите елементи. Така предизвиква неяснота при решаване на проблема със състезанията;
- Логическите изрази изразяват функционалните връзки между входните аргументи, но не изразяват координирано поведение на изходните функции, реализиращи тези отношения. С други думи, булевата логика не е символически завършена логическа система (*symbolically complete*), тъй като тя не съдържа в своя набор такива логически променливи, които да съответстват на нейния край;
- Пълното функциониране на една логическа схема може да се изрази и обясни най-подробно само от нейния създател;
- Булевата логика не може да отрази състезанията, възникващи по време на функциониране на реалните логически схеми и до установяване на истинните логически нива на техните изходи;
- Стига се до извода, че логиката на един булев израз е непълна без допълнителните уточнения (ограничения) от нейния създател.

Авторите на този подход са убедени, че е необходима нова логика, която да допълни булевите оператори с прецизна координация на събитията във времето. Двустойностната нулконвенционална логика (2NCL) на Фант дава решение на повечето от проблемите, свързани с Булевата логика, но както авторът сам признава, нейната реализация неимоверно повишава апаратните разходи. Именно по тази причина, Булевата логика и в частност синхронната се очаква още дълго да доминира в проектирането и производството на електронни схеми.

### 1.2.3. Концепция за “ръкостискане” (*handshake*)

Силно влияние на асинхронния дизайн оказват трудовете на Ivan E. Sutherland, който е предложил трети вариант за самосинхронизиращия се асинхронен модел. Този модел се основава на принципа за конвейерна организация на изчислителния процес. Асинхронният обект на Съдърланд на нивото на интерфейса за връзка с “външния свят” е крайно елементарен и се основава на принципа наречен “ръкостискане” (*handshaking*). По своята същност този принцип изразява обратната връзка в отношенията между два обекта. Принципът на обратната връзка е основен в кибернетиката. В този смисъл управлението се дели на два вида: управление с обратна връзка и програмно управление.

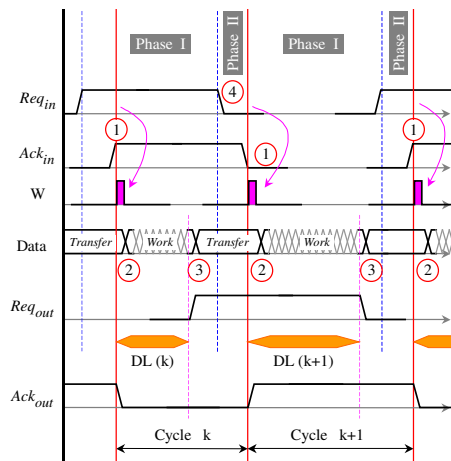
Трябва да поясним, че тук под програмно управление не следва да се разбира изказания от Джон фон Нойман организационен принцип за програмно управление. Последният изразява същността на единството между хардуера и софтуера във всички съвременни компютърни системи. В изказването, което поясняваме, се има предвид, че превключванията в синхронните логически схеми се извършват в отнапред определени моменти, т.е. те са програмирани да се случват чрез тактовия генератор. Променяйки честотата на този генератор, тези превключвания в същност се препрограмират. Следва да отбележим, че такова препрограмиране е неписъщо на асинхронния метод за управление. Въвеждането на асинхронно управление в цифровите системи означава частичен или пълен отказ от програмно управление и преход към управление чрез обратна връзка. Тази концепция се реализира чрез конвейерни автомати, които се синтезират според вида на трансферния протокол.

### 1.3. Протоколи за трансфер

Трансферните протоколи са два вида:

- Двухазов или “без връщане в нула” NRZ (*Non-Return-to-Zero*);
- Четири фазов, или “с връщане в нула” RZ (*Return-to-Zero*);

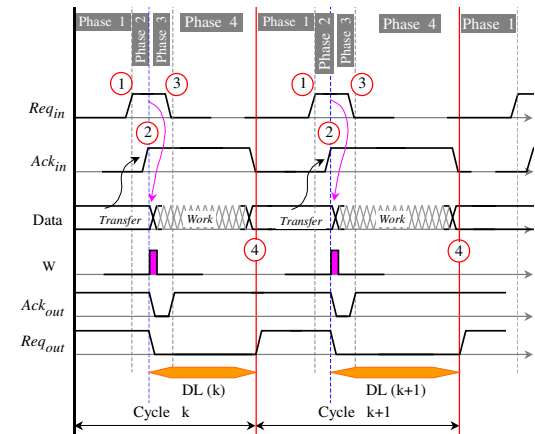
съответно представени на фигура 1.3.2 и на фигура 1.3.3.



Фиг. 1.3.2. Двухазов протокол

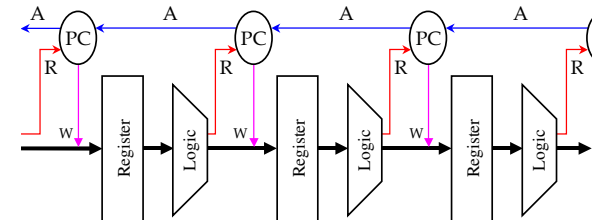
### 1.4. Принцип на конвейерна организация

Последователността на обекти, свързани чрез интерфейса “ръко-стискане”, се определя като микроконвейер. В такава последователност от обекти асинхронният принцип на управление чрез обратните връзки между съседите осигурява по естествен начин конвейерната организация на изчислителния процес. Независимото функциониране на обектите позволява зареждане на нова задача във входния обект винаги когато



Фиг. 1.3.3. Четирифазов протокол

той е готов. Задачите от своя страна се предават от обект към обект, според тяхната готовност. Обектите в микроконвейерите се определят като микроконвейерни звена. Описаната структура е представена на фигура 1.4.1. Задачата на всеки конвейерен автомат PC е да издаде сигнал за запис  $W$  на данни в регистъра фиксатор в нужния момент. Когато той успешно направи това, връща в обратна посока сигнал потвърждение  $A$  (*Acknowledgement*). Всяко микроконвейерно звено генерира сигнал  $R$  (*Request*) заявка за обслужване след като получи резултат.



Фиг. 1.4.1. Структура на асинхронен микроконвейер

В конвейера едновременно във времето могат да се изпълняват толкова задачи от един и същи вид, колкото са степените на конвейера. Всяка задача се намира в отделно звено и изчисленията за всяка една от тях са в различна степен на завършеност. Степента на завършеност съответства на мястото на звеното в последователността, в което се намира.

### 1.5. Закъснителни модели

Конвейерните автомати, които реализират асинхронното управление, използват два вида закъснителни модели:

- Модел на максимално възможно закъснение – *Delay*-модел;
- Модел на фактическото закъснение – *Completion detection*-модел.

## 1.6. Конвейерни автомати

Конвейерният автомат има две задачи:

1. Да следи за онзи най-важен момент, в който той ще може да стартира ново изчисление в микроконвейерното звено, за което отговаря. Най-малкото, което автоматът трябва да направи, е да генерира един фронт, т.е. да се превключи поне веднъж, маркирайки във времето настъпването на това събитие;
2. След като изпълни първата задача автоматът е длъжен веднага да уведоми за това своите съседи, т.е. да разпространи съобщение за събитието, че се е превключил.

Конвейерните автомати се синтезират според вида на трансферния протокол и са два вида – 2-фазови и 4-фазови. Независимо от вида логическата схема на всеки автомат съдържа самосинхронизираща се схема. Обикновено това е С-елемент, чиято схема е представена на фиг. 1.2.1.

### 1.7.1. Мотивация за асинхронен дизайн

Най-съществените мотиви, които насочват към асинхронния дизайн могат да бъдат обобщени така:

- Това е подход, който притежава методология за надеждно проектиране;
- Понякога той е неизбежен и се налага от естеството на събитията (асинхронни интерфейси, арбитражи и др.);
- Придобил е значимост и сила да предопределя избора, тъй като е способен да даде гаранции;
- Не създава проблеми с разпространение на тактови последователности;
- Предразполага и осигурява ниска енергийна консумация;
- Устойчив е към изменения на параметрите (производствени, технологични, температурни, стареене и др.);
- Съвместява хардуерен и софтуерен автоматизиран дизайн.

Асинхронните системи имат потенциални предимства в много аспекти. Техните предимства се проявяват най-вече в посока на:

- Повишена производителност, която се дължи на спестено време от превключване и възможност за конвейерна организация;
- Постигане на ниска енергийна консумация;
- Лекота в прилагането и използването, което се дължи на относителната независимост на отделните блокове;
- Намаляване на зависимостта от електромагнитните смущения.

Асинхронните системи обаче не са универсално приложими. За постигане на предимствата се “заплаща” с увеличени апаратни разходи. Тази обратна страна на този вид дизайн в определени случаи представлява съществен недостатък и силно затруднява избора, ето защо се търси компромис.

## 1.8. Допълнителни аспекти в сравнителния анализ

Сравнителният анализ между проекти, които са свързани със софтуер и такива, които не са свързани, води до следващите обобщения.

### Универсални компютърни системи

1. Програмно управляваните компютърни системи са универсални благодарение на принципа за програмно управление. Такива системи притежават минимално хардуерно осигуряване, в което в даден момент може да се изпълнява една операция.
2. Притежавайки функционално пълна система за минимален набор от хардуерни операционни решения, компютърните процесори използват този ресурс многократно и последователно по силата на софтуерната реализация на съответните алгоритми.
3. Въпреки внедряването на суперскаларност и глобален паралелизъм в компютърните архитектури, тези компютърни системи продължават да бъдат програмно управлявани.
4. Въпреки, че програмно управляваните системи прилагат съвременни хардуерни решения, в това число и асинхронен метод за управление на вътрешно схемно ниво, това не е достатъчно общо, тъй като софтуерната универсалност не им позволява това.

### Хардуерни системи

1. Хардуерните системи не могат да бъдат универсални изчислителни системи. Това се дължи на изключително скъпото хардуерно изменение (разбирано като програмиране или като препрограмиране), което е необходимо за да ги постави в съответствие с произволни алгоритми в различни моменти от времето. Това по същество прави невъзможни на този етап универсалните хардуерни изчислителни системи.
2. В съвременни условия напълно хардуерната реализация може да бъде оправдана само за проекти, които съдържат алгоритми с достатъчно високи изисквания спрямо бързодействието, спрямо производителността, спрямо честотата, с която тяхното изпълнение е необходимо по хода на данния или на изчислителния процес.
3. Хардуерните системи са по-производителни, защото те не изразходват времеви ресурс за обработка на програмен код, реализират в пълна степен икономия на време и по същество са с паралелна форма на организация.
4. Една хардуерна изчислителна система може да се определи като асинхронна когато е синтезирана:
  - според формалния закъснителен модел;
  - според фактическия закъснителен модел;
  - като аперiodично управлявана.
5. Асинхронните хардуерни системи са по-бързи от синхронните, защото изпълняват всяка отделна операция за време, което е фактически необходимо за преработка на конкретно подадените данни. Така хардуерните системи натрупват спестеното време за



всяка отделна операция, изпреварвайки в крайна сметка всяка друга система с равномерен темп на функциониране.

6. Асинхронните хардуерни системи са по-бързи от всички други и в статистически план, тъй като колкото повече пъти (по-често) биват стартирани, толкова повече ще нараства сумата от спестеното за всяко изпълнение време. С други думи, при едни и същи условия, т.е. при фиксиран времеви интервал, асинхронната система ще изпълни заложената алгоритъм повече на брой пъти в сравнение със синхронната система.
7. Асинхронните хардуерни системи са принципно предразположени към конвейерна организация на изпълнявания алгоритъм. Конвейерната организация е най-прилаганата форма на паралелизъм. За хардуерните системи тя може да се определи като тотална, тъй като броят на изпълняваните в нея задачи е равен на броя на конвейерните степени. В този смисъл степента на конвейеризация в програмно управляваните системи е силно ограничена.
8. Все още липсват елементите, които могат да позволят изграждането на хардуерни системи с произволна структура.
9. Опитите да се имплементира в програмируема интегрална схема хардуерен еквивалент на алгоритъм, който е програмно реализиран на език от високо ниво, стигат до реализации, които са далеч от оптималната. Получената реализация при съвременната елементна база е синхронна, а като структура съответства на декомпозиционния модел на В.М. Глушков. Това означава, че управлението се реализира чрез краен автомат с апаратно закрепена логика. По същия начин завършва всеки проект за програмируеми логически интегрални схеми, който е функционално изграждан.
10. Тъй като в синхронните системи надеждното функциониране се осигурява чрез съобразяване с най-бавния елемент, за да се повиши тактовата честота, по време на синтеза се налага да се изпълняват оптимизационни процедури върху отделните съставни елементи на структурата. В същото време в асинхронните системи това не е задължително, тъй като ходът на изчислителния процес в асинхронните вериги не зависи от скоростта на отделните елементи.

### **1.9. Хардуерна реализация на алгоритмични структури**

Въпреки своите недостатъци, елементи на изчислителен процес се реализират хардуерно все по-успешно и тенденцията в това направление е възходяща. За сега автоматизираните средства за проектиране и реализация на синхронни системи са добре развити. Що се отнася до асинхронните системи за тях все още липсват автоматизирани системи. За асинхронните стилове на дизайн се използват наличните автоматизирани системи, но не липсват и радикални подходи, стремящи се да го обезпечат както с подходящ език, така и с технологична среда за проектиране. Достъпните за сега средства правят процеса на проектиране бавен, което го оскъпява и не е предпочитан от производителите.

Основните проблеми, които са обект на научни изследвания, и които съществуват в различни приложни конструкции, бяха вече представени. Това, което обаче нас силно ни интересува, са възможностите за хардуерна реализация на различни алгоритмични структури. Интересът ни към този аспект на проблематиката се мотивира от две главни причини:

1. Изчислителният процес е сложен в структурно отношение, но в същото време е добре изучен. Основните алгоритмични структури, чрез които може да бъде представен произволен алгоритъм са три: линейна, разклонена и циклическа. Всяка една от основните структури е добре позната във всичките ѝ разновидности. В направлението, занимаващо се с анализ и синтез на алгоритми и програми, има натрупан десетилетен опит в изучаването и приложението им. Можем уверено да твърдим, че реализацията на произволен изчислителен процес върху компютърни системи, които по-горе бяха определени като универсални, в съвременни условия не е проблемно;
2. Хардуерната реализация на достатъчно общи алгоритмични структури към сегашния момент не е постигната. Имаме предвид онази хардуерна реализация, която осигурява максимално възможната производителност, която може да осигури асинхронната реализация, прилагайки разнообразни форми на паралелизъм. Възможността за хардуерна реализация на общи алгоритмични структури ще позволи да се "свалят" на по-ниско ниво много от сега софтуерно реализираните изчисления, което в значителна степен би повишило производителността на цифровите изчислителни системи като цяло. Ползността от такива реализации е безспорна и не се нуждае от доказателство.

Литературните източници, които проучваме от гледната точка на асинхронния дизайн обаче в нищожна степен третираат изказания по-горе интерес. Хардуерната реализация на глобални алгоритмични структури все още не е в ползрението на изследванията. Можем да твърдим обаче, че асинхронната линейна алгоритмична структура е обширно и задълбочено изследвана и прилагана. Що се отнася до останалите две (разклонена и циклическа) можем да твърдим, че в проучените от нас литературни източници, не се съдържат изследвания, свързани с тяхната реализация в смисъла, който изказахме по-горе. Времето за това обаче е дошло. От двадесетте дисертационни труда, разработени в последните години, които имахме възможност да проучим, само един съдържа интересен за нас аспект. Изследван е интерфейсен протокол върху конвейерен автомат с условен преход. Анализ на съответната алгоритмична структура обаче не е проведен, не са изявени съпътстващите проблеми и в смисъла, от който ние се интересуваме, изследването е непълно. Освен това изследването не ни изпреварва във времето – факт, който е важен за нас.

В монографиите, които проучвахме, глобална постановка на проблема с хардуерната реализация на общи алгоритмични структури липсва. Липсва структурна декомпозиция на проблема, както и задълбочена анализ, изява и формулиране на задачи и решения.

Основните алгоритмични структури са три. Те образуват функционално пълна система, защото са достатъчни за изразяване на произволен изчислителен процес. Линейната алгоритмична структура е най-елементарната. Хардуерната реализация на тази структура във всички случаи представлява конвейер. Асинхронната ѝ реализация е неоспорим факт.

Условната алгоритмична структура е втората основна структура. В нея изпълнимите елементи изразяват две алтернативни възможности за развитието на изчислителния процес. Тези две възможности са интегрирани чрез условен алгоритмичен преход. Присъствието на двете алтернативни възможности в структурата в същност представлява израз на свойството общност на алгоритмите. За всяко отделно изпълнение ходът на изчислителния процес в такава структура може да бъде уникален, т.е. в съответния момент стойността на кода на условието за преход CJ (*Conditional Jump*) избира един от възможните пътища за хода му в посока към края. И понеже краят е един единствен, разклонената условна структура неизбежно е свързана с обединяване на разклоненията пред входа на общия за тях елемент. Пред входа на общия елемент (в общата точка) всеки клон по същество може да се определи като отделен микроконвейер, така че обединяването на краищата на два или повече конвейера представлява един нов и неизследван проблем. Той е свързан още с това, че алтернативните алгоритмични пътища пораждаат между последователно стартираните в конвейера задачи, състезания.

Ние не сме в състояние да кажем коя от две последователно стартирани в конвейера задачи ще достигне първа общата точка, в случай че същите се разминат по двата алтернативни клона в конвейера. Това е възможно, защото условието за преход е индивидуално за всяка от задачите и стойността му зависи от конкретните данни на всяка задача. В резултат, в общата точка, където всеки от изчислителните процеси ще се бори да продължи своя ход в посока към края неизбежно ще възникне проблемът за избор. Това е нов проблем за асинхронния дизайн на тази алгоритмична структура.

Общата точка, с която неизбежно е свързана условната алгоритмична структура, ще бъде преодоляна от готовите за това процеси. Въпросът е в какъв ред, в каква последователност ще стане това. Не сме в състояние да твърдим, че този ред ще съответства напълно на реда, в който са стартирали задачите. Най-вероятно слизащите от конвейера резултати няма да съответстват по ред на реда на задачите, на които принадлежат. Следователно за следващите изчисления тази разбърканост може да представлява сериозен проблем. Този проблем ние не

откриваме изявен, а още по-малко анализиран и решен в нито един от проучените литературни източници.

В допълнение на това напълно неизследвани са възможностите за хардуерна и конвейерна реализация на многоалтернативните алгоритмични структури, в които присъстват няколко условни алгоритмични прехода и уникалните пътища *begin-end* са много.

Циклическата алгоритмична структура е третата от основните. Тази структура присъства в почти всеки алгоритъм. Говорейки за общи алгоритмични структури тя не може да бъде изключена от вниманието ни. Нейната хардуерна реализация обаче е също напълно неизследвана, тя не е позната в съдържанието на проучваната литература. Очевидно е обаче, че както в разклонените, така и в циклическите структури има алгоритмични преходи, които правят тяхната конвейерна реализация непозната.

Отивайки още по-далече, абсолютно неизследвани в хардуерно отношение са алгоритмичните структури, които могат да се образуват от циклически, като например структури с последователни или с вложени цикли, или цикли в съчетание с условни преходи. Тук не следва да бъдат забравяни структури, които могат да бъдат образувани при реализация и на други форми на паралелизъм. Появата на разклонения например е възможно при реализация на естествен паралелизъм, а така също и при суперскаларност.

В резултат на така изложения анализ става ясно, че хардуерната конвейерна реализация на общи алгоритмични структури е неизследвана област и представлява определено предизвикателство. Ето защо нашето изследване е посветено на формулиране и принципно теоретично решаване на проблемите, свързани с хардуерната реализация на изчислителен процес с обща алгоритмична структура на микрооперационно ниво. Използвайки познати принципи, методи и средства, това изследване е фокусирано върху възможностите за разработване на нови методи за хардуерна реализация на функционално пълна система от алгоритмични структури и е ориентирано преди всичко към асинхронна и конвейерна организация на изчислителния процес. Освен конвейерна организация, ще бъдат търсени реализации и на други форми на паралелизъм. При това ние ще прилагаме синхронния метод на управление във всички случаи, когато той позволява да бъдат получени ефективни решения. В асинхронния дизайн ние възприемаме и ще прилагаме принципа на "ръкостискане".

Направените изводи в настоящия преглед очертават същността на дисертационния проблем като мотивират и обосновават

**целта на дисертационния труд:** разработване в рамките на единен подход на нови методи за синтез на асинхронни микроконвейерни системи с обща структура.

Предприетото изследване цели разработване на методологичната основа за синтез и анализ на хардуерна реализация с неконвенционална организация на микрооперационен изчислителен процес, характеризиращ се с произволна алгоритмична структура. Тя трябва да осигури и да гарантира преди всичко логически правилното функциониране на новите решения и затова ще бъде разработвана с акцент върху качествените аспекти на проблема. Количествените оценки ще бъдат оставени на по-заден план. За постигане на поставената цел трябва да бъдат решени следните **задачи**:

1. Да се анализира възможността за хардуерна реализация на функционално пълна система от алгоритмични структури в условията на конвейерна организация, като алтернатива на микропрограмното управление чрез крайни автомати, с оглед синтеза на нови видове микроконвейерни звена.
2. Да се синтезират и изследват конвейерно организирани реализации на алгоритмични структури с условен преход, с цикъл, с естествен паралелизъм и със суперскаларност. Да се формулират и решат задачите, свързани с хардуерната реализация на тези структури в условията на асинхронна и конвейерна организация на функциониране.
3. Да се анализират микроконвейерни структури, комбиниращи различни видове микроконвейерни звена и различни форми на паралелизъм.
4. Да се изследва възможността операционни схеми да прилагат модела на фактическо закъснение, за постигане на максимално бързодействие.
5. Да се обобщят елементите за асинхронно управление и да се изследва асинхронния подход при синтез на крайни автомати като устройства за управление.

## Глава 2

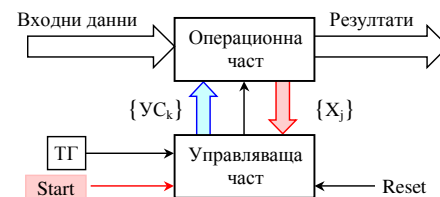
### Микроконвейерни звена

В тази глава е изложено и приложено едно ново разбиране относно възможността за организация на управлението на операционни изчислителни структури. Това разбиране дава възможност за нов подход при създаване на хардуерни устройства и системи, които се характеризира с това, че изключва необходимостта от синтез на крайни автомати за управление. Новата концепция се основава на събитийността и причинно следствените връзки при функциониране на структурите, определени като самосинхронизиращи се. Те от своя страна са изразени в алгоритмите, реализацията на които се търси в тези условия, във вариант, обезпечаващ възможно най-голяма производителност. Високата производителност се постига чрез осигуряване на възможност за естествена реализация на различни форми на паралелизъм, главната от които е конвейерната.

#### 2.1. Операционни структури с управление без крайни автомати.

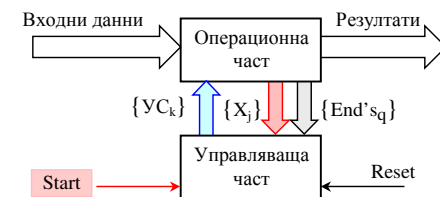
##### Предпоставки за микроконвейерна организация на управлението

Традиционната класическа представа за логическата структура на изчислително устройство е илюстрирана на фигура 2.1.1.



Фиг. 2.1.1. Синхронна структура

Структурата съдържа две основни части – операционна и управляваща според декомпозиционния модел на В. М. Глушков. Операционната част съсредоточава всички логически възли и връзки между тях, които са необходими за преработката на входните данни по точно определен алгоритъм. Управляващата част е синтезирана като краен автомат, който, издавайки в определена последователност управляващи сигнали  $\{UC_k\}$ ,  $k=1,2,3,\dots,M$ , към операционната част, реализира алгоритъма за обработка на данните. По време на управлението, при необходимост, алгоритъмът за управление използва стойностите на множество оповестяващи сигнали  $\{X_j\}$ ,  $j=1,2,3,\dots,N$ . Така управляващата част всъщност съдържа в себе си алгоритъма за управление. Структурата се определя като синхронна, тъй като причината за нейното превключване е тактовия генератор. В случай, че структурата е синтезирана като асинхронна, тя има вида от фигура 2.1.2.



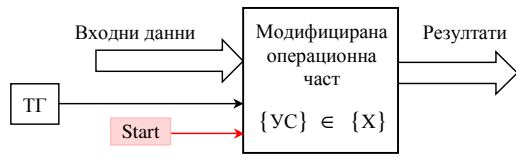
Фиг. 2.1.2. Асинхронна структура

В случая на асинхронно управление тактовият генератор липсва като елемент в структурата за сметка на допълнителното множество сигнали от типа "край на микрооперацията"  $\{End's_q\}$ ,  $q=1,2,3,\dots,L$ .

И в двата метода на управление устройството обикновено се намира в изходно (начално) състояние и изпълнява функциите си след подаване на външен сигнал, формално означен "Start". В изходно състояние то фактически очаква този сигнал.

В класическите реализации управляващата част на структурата реализира управляващия алгоритъм на ниво микрооперации като краен автомат с апаратно закрепена логика или като краен автомат с програмно закрепена логика.

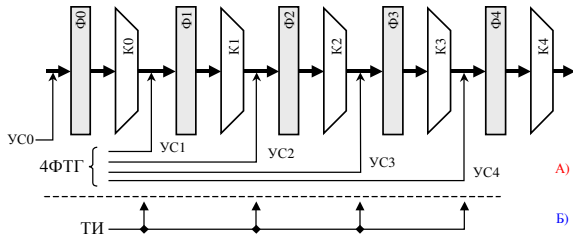
Новата концепция, предложена в дисертационния труд, се основава на увереността, че основните алгоритмични структури, които могат да се съдържат в управляващия алгоритъм, и които се реализират технически (апаратно или микропрограмно) в управляващата част на устройството, могат да бъдат прехвърлени (вградени) в операционната му част. Формално това може да се изрази чрез принадлежността на управляващите сигнали към множеството на оповестяващите сигнали. С други думи, основавайки се на причинно следствените връзки, имаме право да разглеждаме управляващите импулси, като причинители на превключванията и възникващите в резултат на това признаци, като сигнали на едно и също множество. При постигане на тази цел, в изчислителната структура остава единствено модифицираната операционна част, получена в резултат на съответното преобразуване на управляващата част.



Фиг. 2.1.3. Структурата на устройство без автоматно управление

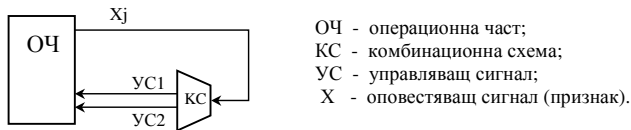
По същество тази нова концепция е обоснована върху трите основни алгоритмични структури:

- Линейна, която може да се синтезира като линеен синхронен или асинхронен микроконвейер, във вида, показан на фигура 2.1.7;



Фиг. 2.1.7. Линейна конвейерна структура

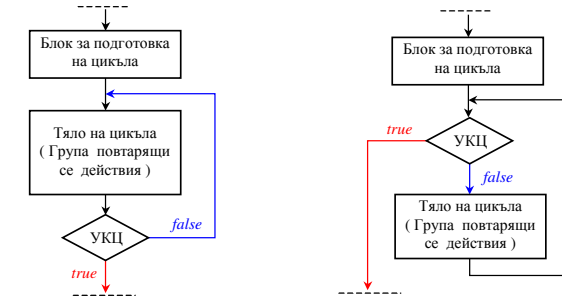
- Условна. Разбирането за апаратната реализация на управлението в този случай е за самоопределяне на микрооперацията, илюстрирано на фигура 2.1.10;



Фиг. 2.1.10. Самоопределяне на микрооперацията

Самоопределянето води до синтез на два микроконвейерни клона в операционната част на устройството.

- Циклическа. Циклическите структури съдържат условен алгоритмичен преход.

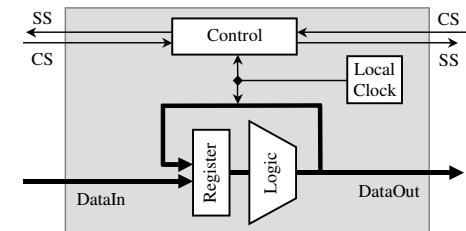


Фиг. 2.1.12. Циклическа алгоритмична структура

Циклически структури с известен брой повторения могат принципно да бъдат сведени до линейни чрез разгъване. За структури с неизвестен брой повторения това не е възможно. Налага се пълно изследване на този вид структури.

## 2.2. Синхронни многотактови микроконвейерни звена

Циклическата алгоритмична структура, която е синтезирана и изследвана в различни условия, е дефинирана като многотактово микроконвейерно звено чрез общия вид на структурата, представена на фигура 2.2.1.



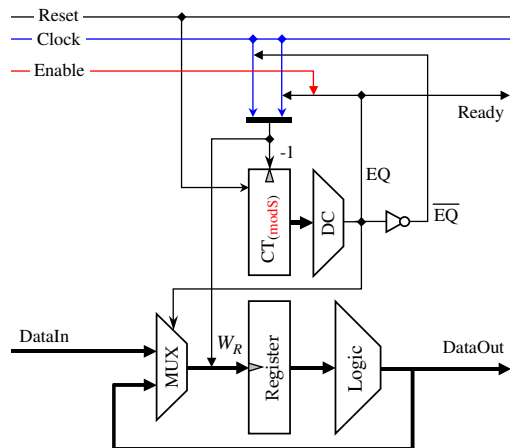
Фиг. 2.2.1. Обобщена структура на многотактово МКЗ

Структурата съдържа три основни елемента – конвейерен регистър фиксатор (*Register*) и съвкупност от комбинационни логически схеми (*Logic*), реализиращи необходимата преработка на данните. Най-същественото в тази структура е вътрешната обратна връзка.

В следващите раздели (от 2.2 до 2.9) на тази глава са представени различни решения за циклическата алгоритмична структура във вид на синхронни многотактови микроконвейерни звена. По разбираеми причини някои от тях ще бъдат само споменати.

### 2.2.1. Конвейерно звено за цикъл с известен брой повторения

Синтезираното в този раздел микроконвейерно звено има логическата структура, представена на фигура 2.2.2.



Фиг. 2.2.2. Многотактово микроконвейерно звено с брояч по модул  $S$

Структурата е тактувана само от едната фаза на сигнала. Тя притежава асинхронен вход, чрез който в нея се стартират изчисленията. Тя е синтезирана върху времевите причинно следствени събития и не притежава отделна явно изразена управляваща част. Тя е универсално синхронно микроконвейерно звено и може да се вгражда в синхронни и асинхронни микроконвейери.

### 2.2.2. Конвейерни звена за цикъл с променлив брой повторения

Синтезираното микроконвейерно звено има аналогична логическа структура, в която към брояча е изграден допълнителен даннов вход, което я прави програмируема. Така в нея могат да се изпълняват цикли с известен брой повторения, който обаче може да бъде променян. Началното съдържание на брояча се записва заедно с това на регистъра фиксатор. Структурата притежава изход, който показва състоянието й.

За този вид звена са синтезирани още варианти за циклически структури с предусловие.

### 2.3. Синхронни многотактови звена, тактувани с един и същи фронт

В този раздел е представен синтезът на две логически структури на многотактови звена за цикли с известен брой повторения и на програмируеми такива. Синтезът е извършен при условието, че логическите възли ще използват само преден фронт на управляващите импулси, условие, което в много случаи налага фамилияте от реални гравивни елементи.

Условието, което поставят гравивните елементи налага използването и на двете фази на тактовата последователност. Това се отразява в някои особености на функционирането, изразяващи се в появата на състезания в обратните връзки на управляващите схеми. Въпреки, че това не пречи на функционирането, налагат се допълнителни ограничения.

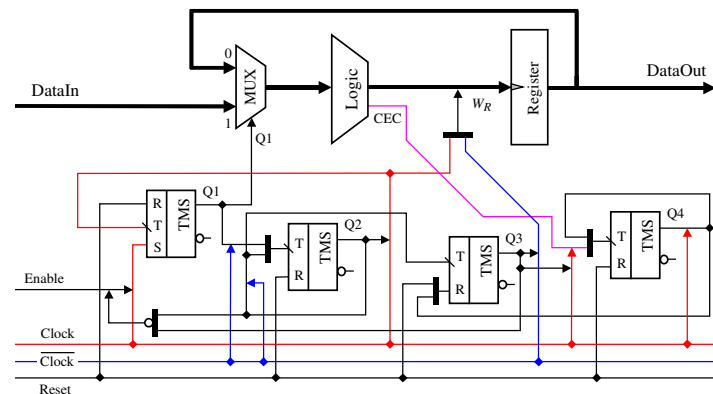
### 2.4. Програмируемо многотактово звено с цифров компаратор

За фамилии гравивни елементи, които не предлагат декрементни броячи е синтезирана програмируема логическа структура на многотактово звено с цифров компаратор. Однофазното тактуване налага използването на регистри, превключващи се по различни фронтове.

### 2.5. Многотактово звено с неизвестен брой повторения

Алгоритмичната структура на цикъл с неизвестен брой повторения е структура, която най-трудно се поддава на модифициране. Условието за край на повторенията се формулира от същността на изчисленията.

Структурата на многотактовото звено е синтезирана въз основа на времевите причинно следствени събития и не притежава отделна явно изразена управляваща част.

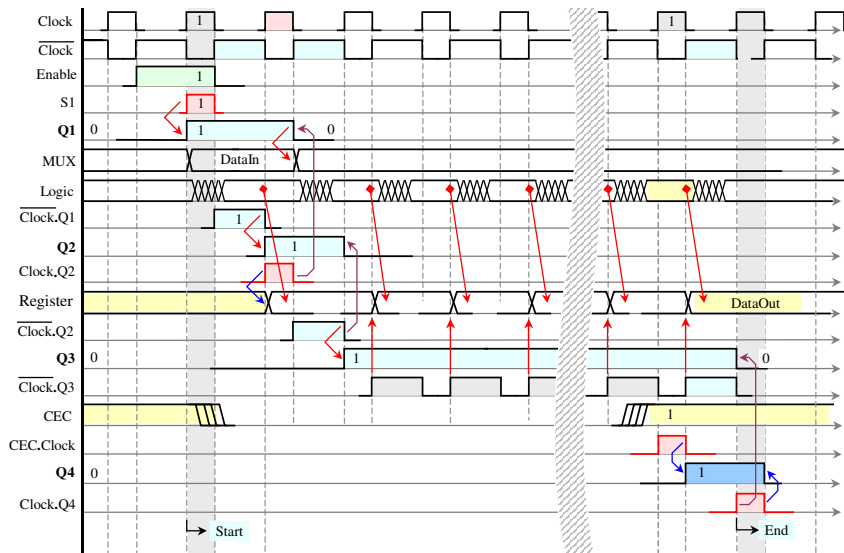


Фиг. 2.5.1. Циклическо звено с неизвестен брой на повторенията

Условието за край на циклическите повторения CEC в тази структура се изчислява от операционната логика на звеното при всяка итерация. Структурата е тактувана от двете фази на тактовата последователност. Тя е универсално синхронно микроконвейерно звено и може да се вгражда както в синхронни така и в асинхронни микроконвейери. Време-диаграмата за функциониране е показана на фигура 2.5.2.

### 2.6. Микроконвейерни звена без вътрешни състезания

При синтеза на синхронните многотактови звена като основен проблем на преден план излизат разнообразните постановки на вътрешната



Фиг. 2.5.2. Времедиаграма на циклическия процес

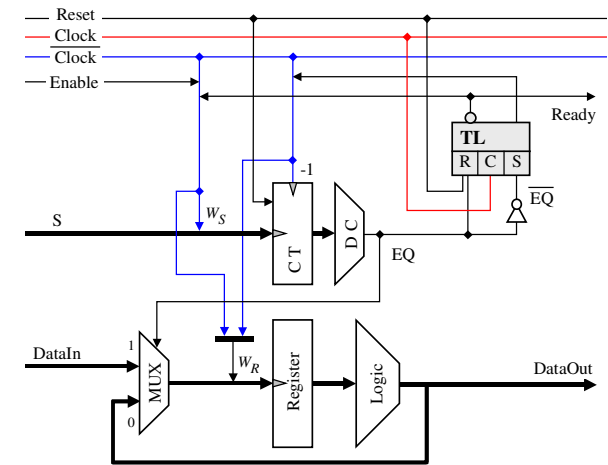
синхронизация на структурите. Това налага използването на логически възли с различно превключване или използването на двете фази на тактовата последователност. Ограниченията от гравивните елементи налага максимално ефективно използването на параметрите на тактовата последователност. В настоящия раздел е предложено ново решение за две такива логически структури, с помощта на което се постига синхронизация на управлението и отстраняване на вътрешните състезания. Синтезът включва в структурата допълнителен синхронизиращ тригер. Синтезирани са две структури за многотактови звена с известен брой повторения. На фигура 2.6.4 е представена програмируемата структура на звеното, а на фигура 2.6.5 времедиаграмата му.

### 2.7. Многотактово микроконвейерно звено с два регистъра

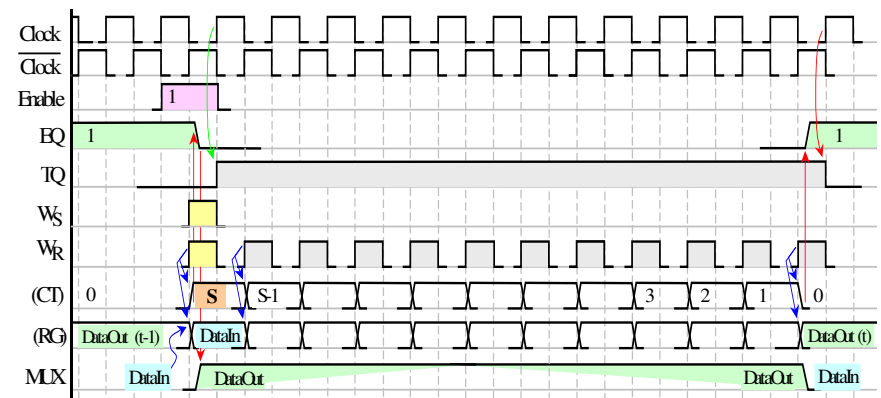
Представеното тук микроконвейерно звено е разработено като алтернативен метод на изложения в предходен раздел такъв и се отнася за циклически алгоритмични структури с предварително неизвестен брой повторения.

### 2.8. Микроконвейерно звено с многотактово тяло за цикъл с известен брой повторения

Синтезът на многотактово звено, чието циклическо тяло е многотактово тук е съчетан с конвейерна организация на тялото. Това се изразява в зареждането му с пакет от няколко задачи, зареждани през входната му точка. Пакетът от задачи се превърта в тялото многократно. Зада-

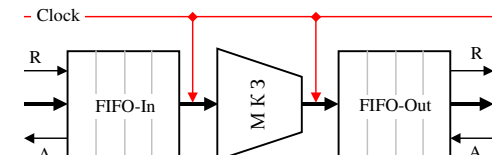


Фиг. 2.6.4. Програмируема структура, работеща по преден фронт



Фиг. 2.6.5. Времедиаграма на цикъл

чите напускат конвейерното тяло на цикъла последователно заедно с ново зареждания пакет от задачи. Синхронизацията на структурата като микроконвейерно звено в състава на конвейер изисква добавянето на FIFO-буфери в структурата, показана на фигура 2.8.3.



Фиг. 2.8.3. Общ вид на структурата на звеното

Интерфейсът на звеното включва сигналите (R) *Request* и (A) *Acknowledgement*.

### 2.9. Микроконвейерно звено с многотактово тяло за цикъл с неизвестен брой повторения

Видът на този цикъл създава най-съществените затруднения в синтеза на тази конвейерна структура. Най-същественото, което ще споменем тук се отнася до функционирането. То се изразява в това, че при напускане на цикъла, редът на задачите не съответства на реда, в който са били зареждани. Освен това повторно зареждане на структурата със нов пакет задачи не е възможно докато и последната от предишния пакет не напусне циклическото тяло.

## Глава 3

### Асинхронни микроконвейери

В тази глава като цяло са изложени проблемите, свързани със синтеза на общи и уникални по структура асинхронни микроконвейери и микроконвейерни системи. Анализът на всяка от разгледаните теми изяснява множество нови научни задачи. Такива са задачата за възстановяване на реда на конвейерните резултати, задачата за арбитражиране на заявките в общите точки, задачата за естествен паралелизъм, за суперскаларност, както и задачите за синтез на множество оригинални конвейерни автомати. Представени са линейни и нелинейни конвейери, конвейери с циклически структури, паралелни и суперскаларни конвейери. Асинхронната конвейерна организация се осъществява чрез принципа на “ръкостискане” между съседни микроконвейерни звена. Този принцип се реализира чрез конвейерните автомати. За нуждите на тези конвейери са синтезирани и изследвани множество нови и оригинални логически схеми на конвейерни автомати.

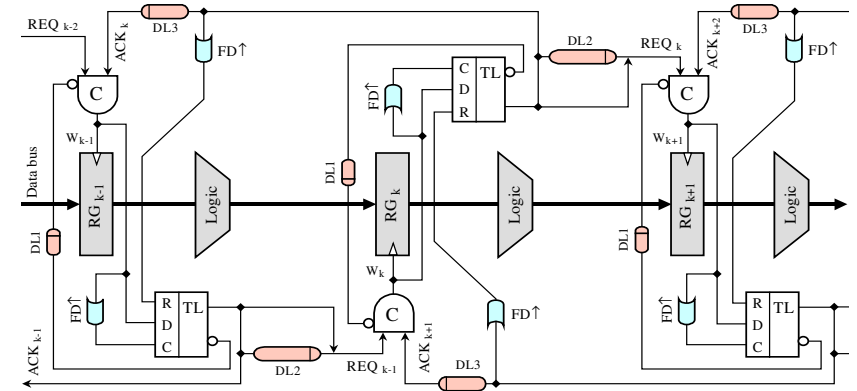
Изложеното в глава 2 налага актуализиране на структурата на многотактовото звено. Освен това тук е направена нова интерпретация на структурата, която извежда на преден план микрооперация запис в конвейерния регистър фиксатор, което е важно за събитийното управление на конвейер, комбиниращ еднотактови и многотактови звена.

### 3.1. Четирифазов асинхронен микроконвейер с еднотактови и многотактови микроконвейерни звена

След отчитане на изложените съображения както относно протокола, така и относно регистрите фиксатори, е синтезиран вариант на структурната схема на микроконвейер, в който могат да се редуват както еднотактови така и многотактови микроконвейерни звена. Схемата е представена на фигура 3.1.5.

Синтезираният конвейерен автомат (PC, *Pipeline Controller*), чрез който е реализиран линейният конвейер съдържа *D-Latch* тригер. Протоколът на интерфейса е 4-фазов с изпреварващо нулиране.

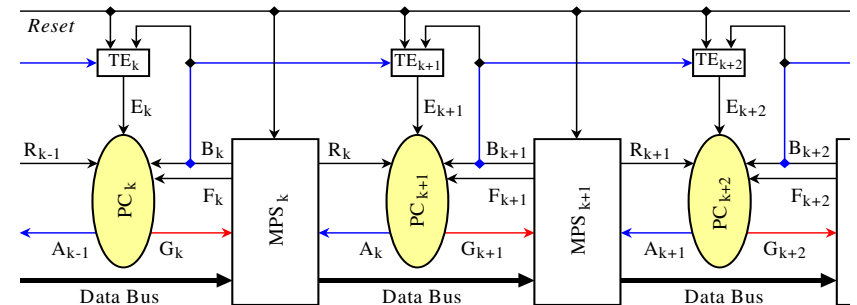
Предложен е втори вариант на такъв конвейер, за който синтезираният конвейерен автомат съдържа *D-Edge* тригер. Фигура 3.1.9 представя реалното включване на многотактово звено заедно със синхронизатора и конвейерния автомат. Вижда се реалното използване на статус сигнала на звеното в качеството му на модел на фактическо закъснение.



Фиг. 3.1.5. Структура на микроконвейера

### 3.2. Асинхронен микроконвейер с многотактови звена

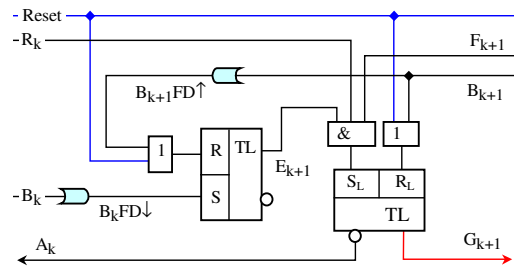
Конвейерът представен в този раздел (фигура 3.2.1) е линеен от многотактови звена, които имат интерфейс, съдържащ естествени статус-сигнали “Свободен” (F), “Зает” (B), “Готов” (R).



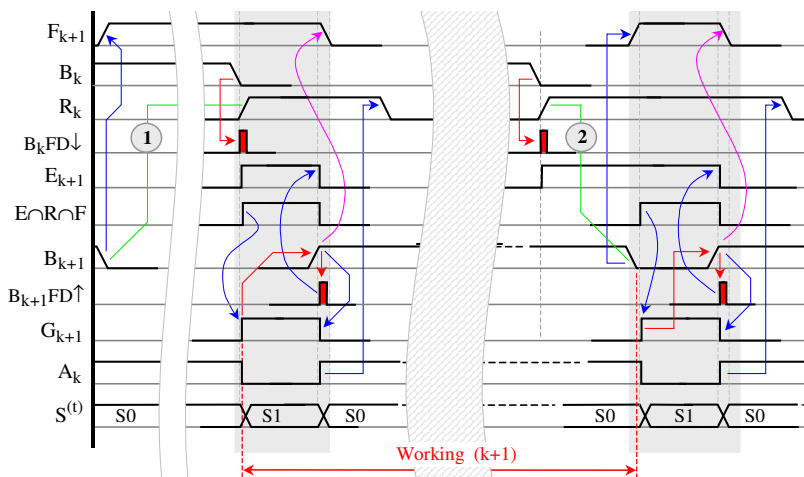
Фиг. 3.2.1. Микроконвейер с многотактови звена

За този интерфейс е синтезиран конвейерен автомат и 4-фазов протокол с изпреварващо нулиране – фигура 3.2.3.

Преходният процес, който съответства на графа на автоматата, е изобразен два пъти – в лявата и в дясната част на времедиagramата от фигура 3.2.4 и представя стартирането, работата, спирането и повторното стартиране на микроконвейерно звено №(k+1).



Фиг. 3.2.3. Логическа схема на конвейерния автомат



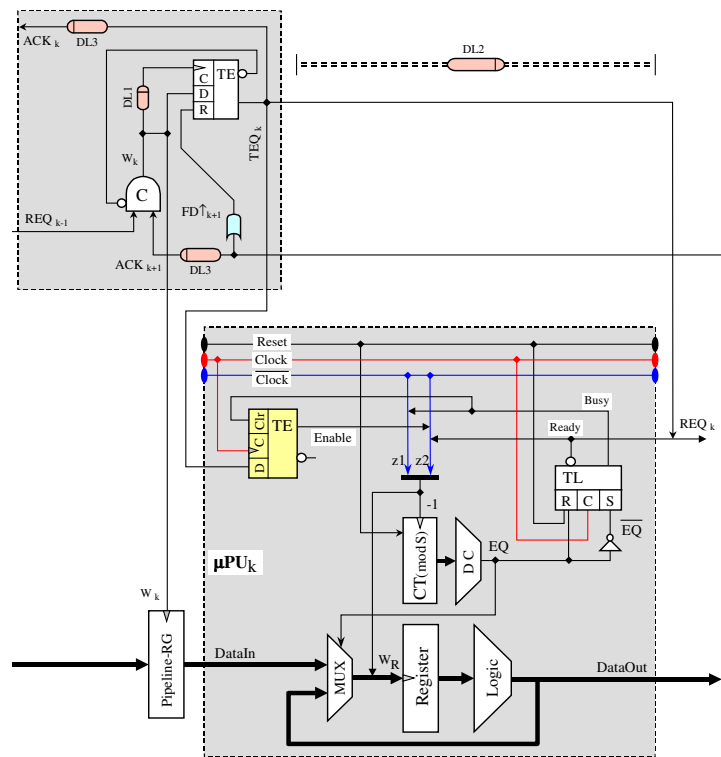
Фиг. 3.2.4. Времева диаграма на трансферния протокол

### 3.3. Нелинейни асинхронни микроконвейери

В този раздел са разработени проблеми, породени от включването в хардуерната структура на конвейерите на условни алгоритмични преходи. В структурата на такива конвейери се съдържат паралелни конвейерни участъци, което ги определя като нелинейни. Анализът на тази алгоритмична структура формулира 4 нови научни задачи:

1. Задача за синтез на конвейерен автомат в точката на условния преход. Този автомат съществено се различава от обикновения линеен автомат по това, че той трябва да реши задачата за избор на един от двата алгоритмични клона. Тази задача е неизбежно свързана със следващите. Всички нови задачи са свързани със синтез на конвейерни автомати в особените точки на алгоритмичните структури от общ вид.

Апаратната реализация на микроконвейерно звено с условен алгоритмичен преход беше разгледана в раздел 2.1, където е изложен анализът на проблема и са предложени два варианта за неговата реализация.



Фиг. 3.1.9. Свързване на конвейерен автомат с многотактово звено

Общото между тях е това, че стойността на логическото условие за преход CJ се използва в звеното за управление на данновата шина, за да бъдат резултатите насочени към актуалното разклонение, реализирани по този начин конкретен алгоритмичен път *Begin-End*. Логическата стойност на това условие трябва да реши в кой от клоновете трябва да постъпи заявката *Request*. С тази задача е свързана задачата за приемане на сигнала потвърждение *Acknowledgement* от съответното звено, приело заявката. Поради неразривната същност на сигналите *Req* и *Ack* ние приемаме задачата за една, въпреки двете отделни решения, които тя притежава.

2. Задача за синтез на конвейерен автомат, който ще управлява микроконвейерни звена в общите точки на алгоритъма. Постъпването на входа на дадено звено на готови резултати, идващи от няколко предходни звена, разположени топологически паралелно, се изяснява като нов и самостоятелен проблем. Същността на неговото решение, се състои в синтез на конвейерен автомат, управляващ приемащото микроконвейерно звено при паралелно постъпване във времето на повече от една

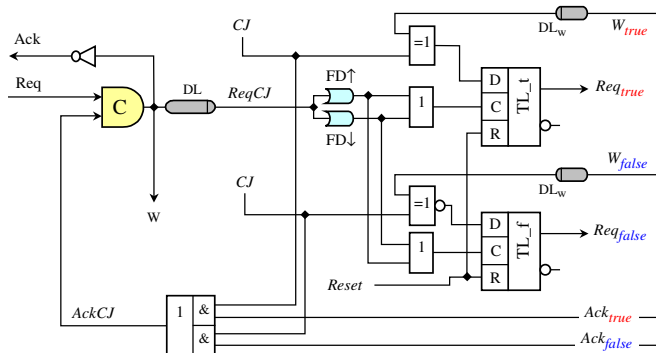


заявка  $Req$ . В огледален смисъл с тази задача е свързана още една задача – задачата за генериране на сигнал потвърждение  $Ack$  към съответно свързаното предходно звено. Поради неразривната същност на сигналите  $Req$  и  $Ack$  и тук приемаме задачата за една, но тя също ще прилежава две отделни решения.

**3.** Задача за избор на заявка. За решаване на задачата за избор на заявка и приемане на съответните данни, конвейерният автомат на приемащото звено трябва да изпълни арбитражна процедура.

**4.** Задача за възстановяване на реда на слизащите от конвейера резултати. В общата точка, където се обединяват няколко разклонения, заявките, които придружават получените в тези клонове данни, се отнасят до различни задачи, стартирани преди това в конвейера. Редът, в който съответните им междинни резултати достигат точката на обединяване, съвсем не може да се очаква да съответства напълно на реда, в който са стартирани задачите, на които те принадлежат. С други думи, в приемащото звено, което стои в общата точка, едва ли ще постъпват данни в правилния ред. Така наличието на разклонения в конвейера води до проблем, същността на който се състои в това, че излизащите на изхода на конвейера резултати едва ли ще бъдат в онзи ред, който съответства на реда на стартиране на задачите, на които принадлежат.

Освен формулирането на горните задачи в този раздел е представено решение на Задача 1. Синтезирани са два варианта: автомат с 2-фазов и с 4-фазов протокол за звено с условен преход. На фигура 3.3.3 е показана схемата на 2-фазовия конвейерен автомат.



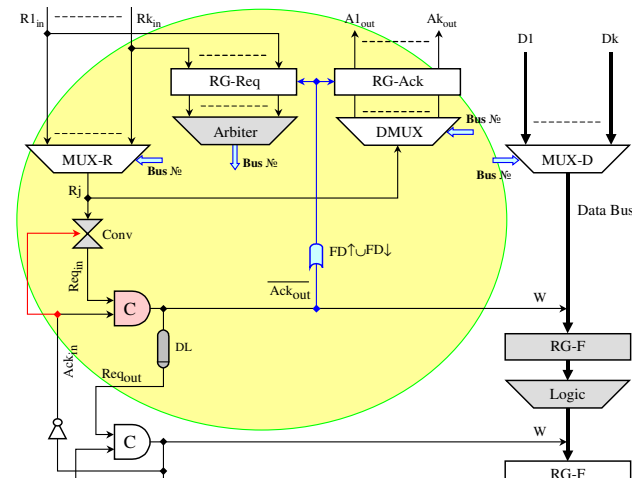
Фиг. 3.3.3. Схемата на 2-фазовия автомат за звено с условен преход

### 3.4. Управление на звено в обща точка на микроконвейер

В този раздел е представено решението на Задачи 2 и 3. Конвейерният автомат в общата точка на разклонена структура има много входове за заявки и много изходи за сигнали, връщащи потвърждение. Той прави избор измежду звената, подаващи в текущия момент заявки.

В основата на този избор като справедлива е положена следната стратегия: заявките ще се избират последователно в кръг, при конструктивно заложен ред. Редът може да съответства на естественото номериране на клоновете (1, 2, ..., k), което условно ще направи конструктора. Тази стратегия гарантира обслужване на заявка от произволен клон след не повече от (k-1) на брой предхождащи обслужвания на заявки от останалите клонове. Стратегията се реализира чрез неактивна схема за избор, която е в състава на синтезирания асинхронен арбитър.

Както за предидущата задача, така и за тази са синтезирани два варианта на конвейерен автомат за звено в обща точка: с 2-фазов и с 4-фазов протокол. На фигура 3.4.4 е представена схемата на 2-фазовия автомат.



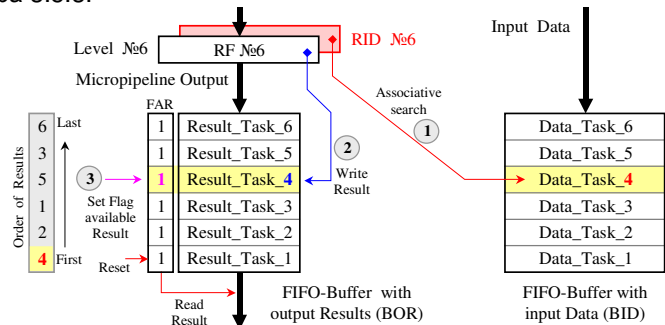
Фиг. 3.4.4. 2-фазов конвейерен автомат в обща точка

Получените от отделните клонове заявки се регистрират в RG-Req. Арбитърът, който е решение на Задача 3, избира една и чрез получения код за избор управлява превключването на останалите възли в структурата на автомата. След данновия трансфер между избраното звено и звеното в общата точка, обслужената заявка се унищожава, за да не пречи на следващия избор.

### 3.5. Възстановяване на реда на резултатите, слизащи от конвейер

В този раздел е представено решението на Задача 4. Причина за разбъркването са условните преходи в алгоритъма, които определят в него няколко алтернативни пътя *Begin-End*. Конвейерната организация е другата причина. Всяка от заредените в конвейера задачи има уникален алгоритмичен път, което е предпоставка за състезания. Решението на проблема се основава на идентификатор на отделните задачи, на оригинална структура на конвейерен буфер и на оригинален алгоритъм за

неговото управление. Общият вид на конвейерния буфер е представен на фигура 3.5.3.



Фиг. 3.5.3. Структура на микроконвейерния буфер

В него се съдържат два отделни FIFO-буфера. BOR – буфер за изходните резултати и BID – за входните идентификатори. За такива са избрани входните данни на всяка задача, които се придвижват в конвейера заедно с нея. Всяка клетка в BOR има по още един бит FAR (*Flag for Available Result*), в който чрез 1 се маркира факта, че в нея е записан резултат.

Алгоритъмът за възстановяване на реда изразява по същество операциите запис и четене, извършвани в конвейерния буфер. Тъй като той представлява звено в конвейера, първата се изпълнява от конвейерния автомат на неговия вход, а втората от автомата на изхода. Всяка заредена в конвейера задача записва входните си данни в BID. Те са нейния идентификатор ID и съпровождат задачата при нейното движение в конвейера чрез регистрите RID. Когато в изходното звено на конвейера се получи резултат, в BID се извършва асоциативно търсене по ID. Клетката, генерираща при това търсене съвпадение, определя къде в BOR трябва да бъде записан този резултат. След запис на резултата в така достъпната клетка, същата се маркира и FAR:=1. Ако този флаг маркира изходната клетка в буфера BOR, нейното съдържание може да бъде прочетено, след което и двата буфера BOR и BID преместват данните си в посока към изхода.

Както се разбира алгоритъмът се основава на това, че в BID се поддържа правилния ред, който чрез асоциативно търсене, определя мястото на поредния резултат в BOR. В този раздел подробно са обяснени операциите запис и четене, които буферът изпълнява, както всички особени ситуации, които могат да настъпят.

Програмно е моделиран конвейерен буфер за примерен алгоритъм.

Чрез програмния модел числено е експериментирана функционалната връзка между обема на буфер и броя на степените в конвейера. Установено е, че производителността на конвейерната система достига максимум при обем равен на броя на конвейерните звена.

### 3.6. Автомати за управление на звена в общи точки на конвейери, съдържащи буфер за възстановяване реда на изходните резултати

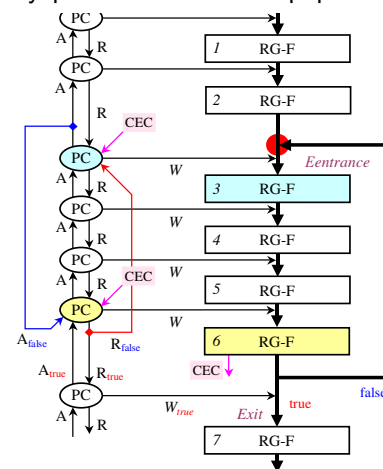
В този раздел се разглежда система от конвейер и конвейерен буфер за възстановяване на реда. Анализът доказва, че функционирането на системата не е застрашено, когато арбитърните схеми в конвейерните автомати, управляващи микроконвейерните звена в общите точки на структурата не откриват фиксирани заявки. Арбитърните схеми могат да се окажат в състояние на очакване, ако се предхождат от микроконвейерни звена с голяма латентност. Това са случаи, когато възлово звено блокира хода на задачите и докато не завърши изчисленията си конвейерната система остава в привидно статично състояние.

### 3.7. Автомати за управление на звена в общи точки на конвейерни системи и разгнати цикли

В този раздел е показано, че когато конвейерната система може да бъде синтезирана чрез използване подхода на пълно разгъване на цикли, новите характеристики на конвейерната система са положителни. За повишената производителност обаче се “заплаща” със допълнителни апаратни разходи. Анализът показва, че разгъването на цикли е възможно само когато са от вида с известен брой повторения. Благодарение на увеличени брой степени в конвейера се повишава производителността и се подобрява ритмичността, в резултат на което се отстранява статичното състояние на системата.

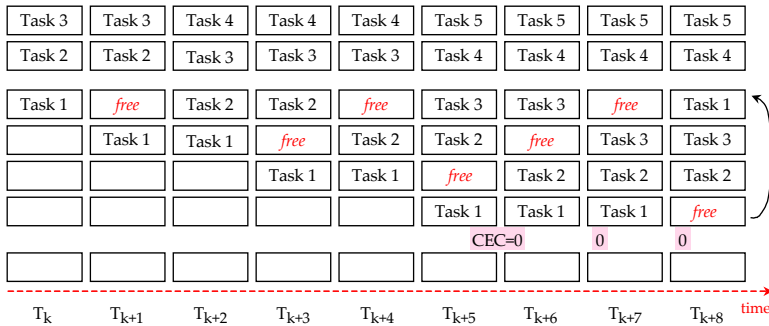
### 3.8. Асинхронни циклически микроконвейерни структури с многотактови тела със следусловие

В този раздел е разгледана циклическа алгоритмична структура с многотактово линейно тяло, която е асинхронно и конвейерно организирана. Общият вид на управлението е илюстрирано с фигура 3.8.2.



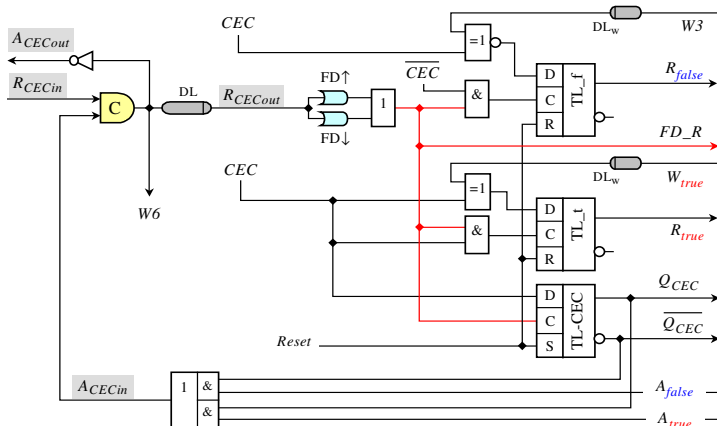
Фиг. 3.8.2. Асинхронна конвейерна организация на цикъл

В циклическата структура има две съществени точки. Това са входната точка (*Entrance*) и изходната точка (*Exit*). Управлението на микроконвейерните звена в тези две точки е в зависимост от логическата стойност на условието за край *CEC*. Конвейерните автомати на входното и на изходното звено остават свързани през цялото време на циклическото изчисление. Както и във синхронния вариант и тук циклическото тяло трябва да бъде първоначално заредено с пакет задачи – процес, който е илюстриран на фигура 3.8.3.

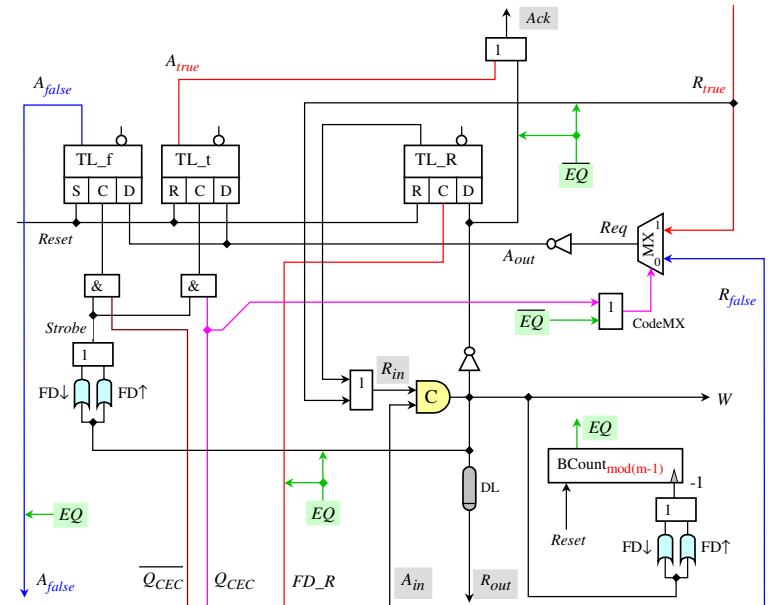


Фиг. 3.8.3. Навлизане на много задачи в цикъла и превъртане

В този раздел са разгледани два вида цикли – с известен и с неизвестен брой повторения. За постигане на преследваната организация на функциониране са синтезирани принципните логически схеми на конвейерни автомати, управляващи звената във входната и в изходната точки на циклическата структура. Синтезирани са конвейерни автомати за 2-фазов и 4-фазов трансферни протоколи. На фигура 3.8.8 е представен 2-фазовият вариант, а на фигура 3.8.14 е представен 4-фазовия вариант на логическата схема на автомата в изходната точка на цикъла.

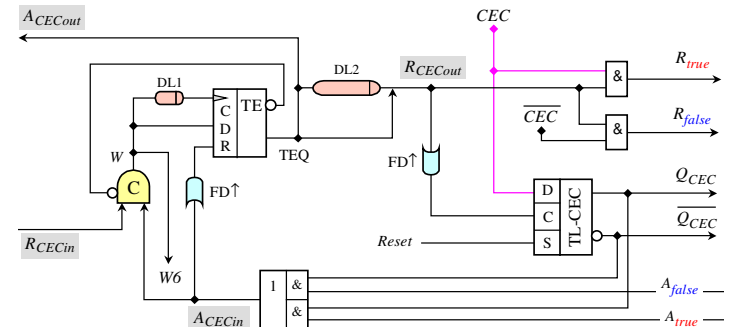


Фиг. 3.8.8. Схема на 2-фазов автомат за звено с условие за край на цикъл



Фиг. 3.8.13. Схема на 2-фазов автомат за входно звено на цикъл

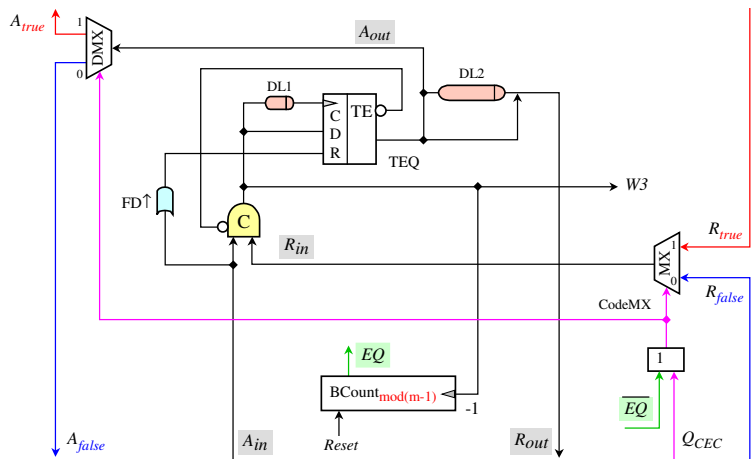
Конвейерният автомат във входната точка се управлява при първоначално зареждане отделно, а от условието *CEC* се управлява, след като последното получи първата си актуална стойност. И за този автомат са синтезирани две логически схеми – съответно за двата вида протоколи. Схемите са представени на фигури 3.8.13 и 3.8.16.



Фиг. 3.8.14. Схема на 4-фазов автомат за изходно звено

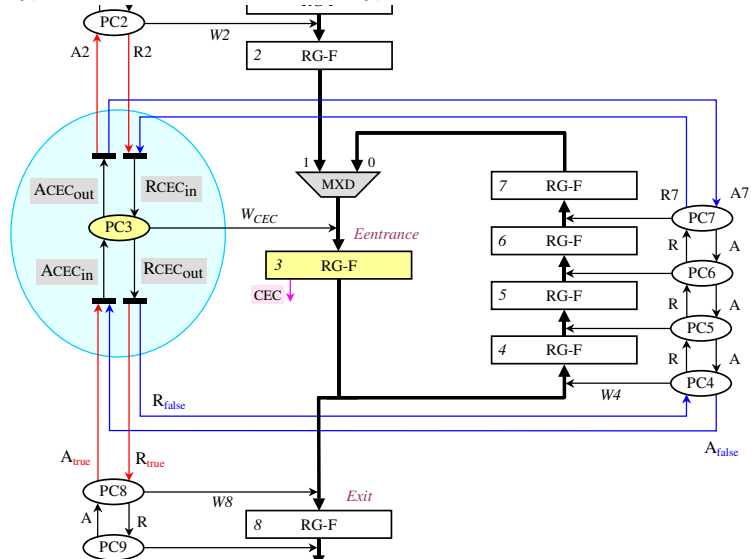
### 3.9. Асинхронни циклически микроконвейерни структури с многотактови тела с предусловие

В този раздел се разглежда задачата за конвейерна организация на циклическа алгоритмична структура с предусловие. Най-характерното за



Фиг. 3.8.16. Схема на 4-фазов автомат за входно звено на цикъл

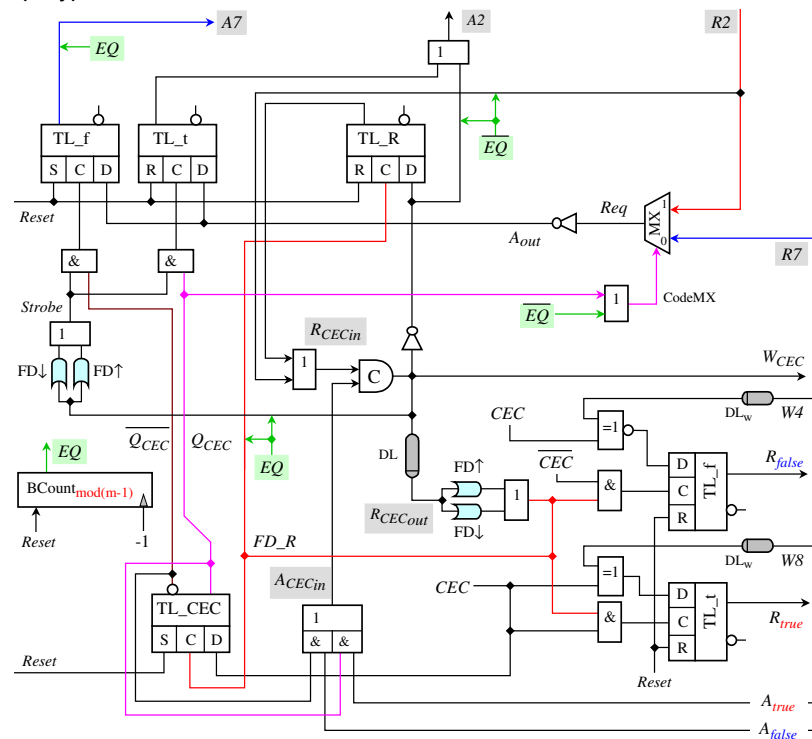
нея е това, че тя обединява входната и изходната точка на циклическата структура в една точка. В резултат на това конвейерната циклическа структура има вида, показан на фигура 3.9.1.



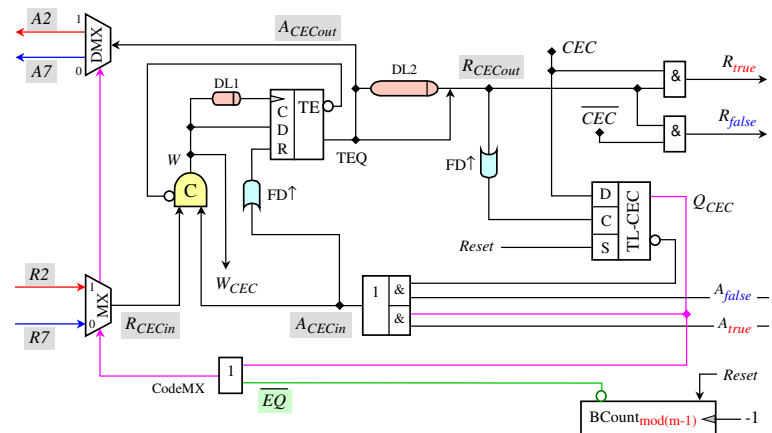
Фиг. 3.9.1. Структура на асинхронен микроконвейер за цикъл с предусловие

Това обединение води до обединение и на конвейерните автомати. В крайна сметка за управление на микроконвейерното звено в общата точка е синтезиран конвейерен автомат в два варианта – за 2-фазов и за 4-фазов трансферен протокол, представени съответно на фигура 3.9.4 и

на фигура 3.9.7.



Фиг. 3.9.4. Логическа схема на 2-фазов конвейерен автомат



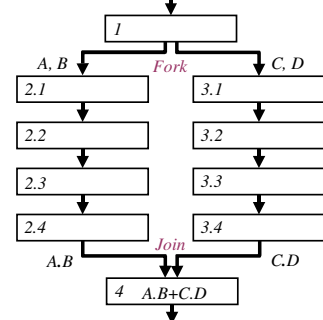
Фиг. 3.9.7. Логическа схема на 4-фазов конвейерен автомат

### 3.10. Общо разположение на циклически тела

В този раздел е направен анализ на възможните комбинации от циклически конвейерни структури, като последователни цикли, вложени цикли или разклонения с цикли. Показано е, че реализацията на такива структури не изявява нови задачи, които да са свързани със синтез на нови по вид конвейерни автомати. Това се дължи факта, че в тези структури не се съдържат други по вид точки, различни от вече разработените. Това са точки с условни преходи, общи точки, входни точки и изходни точки. Новите проблеми, които пораждаат такива комбинации са свързани с общата синхронизация на отделните структурни елементи в конвейерите. Показано е, че общата синхронизация, в смисъла на равномерен темп на функциониране, е затруднена, което се дължи на различните параметри на циклическите структури. Тези проблеми се оказват много сериозни и изискват допълнителни изследвания, излизащи извън темата на дисертационния труд.

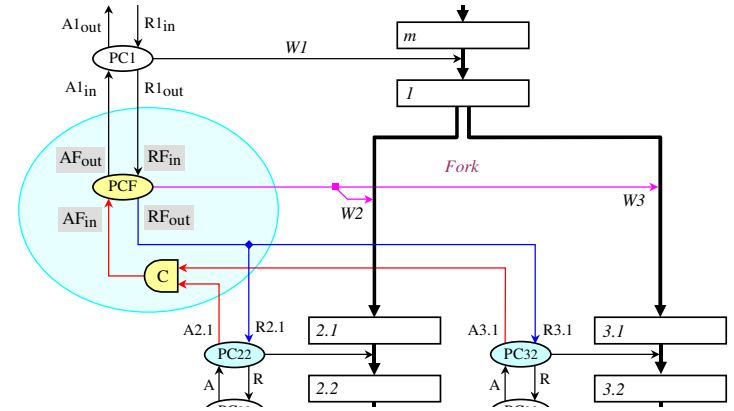
### 3.11. Паралелни микроконвейери

В този раздел е представена конвейерната реализация на естествения паралелизъм – форма, която най-често се съдържа в математическите изрази. Декомпозицията на математическите изрази може да осигури възможност за паралелно изчисление на получените подизрази с последващо обединяване в рамките на последната операция. Конвейерната организация на тази форма на паралелизъм води до едновременно стартиране на изчисления в паралелните разклонения и обединяване на техните крайни резултати. Получената конвейерна структура (фигура 3.11.1) има паралелно разположени конвейерни клонове, функциониращи под общо управление.



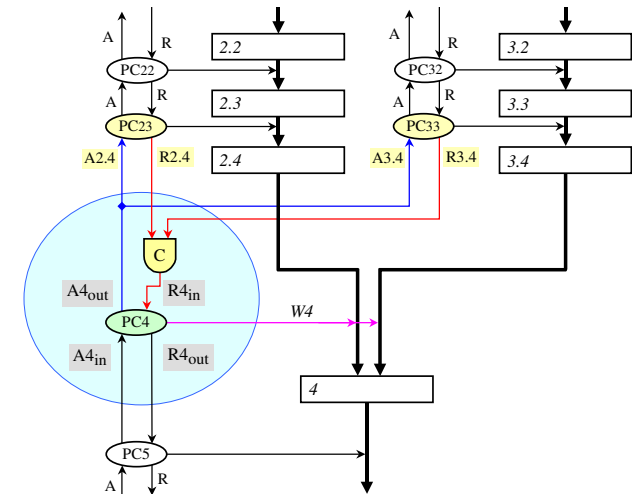
Фиг. 3.11.1. Естествен паралелизъм

За управление на звената в точката на разклонение (*Fork*) е синтезиран конвейерен автомат, показан на фигура 3.11.2. Логическата схема на автомата PCF (*Pipeline Controller into Fork dot*) има два варианта, съответно за двата вида трансферен протокол.



Фиг. 3.11.2. PCF - конвейерен автомат в точката на разклонение

За управление на звената в общата точка (*Join*) е синтезиран конвейерен автомат, показан на фигура 3.11.6. Логическата схема на автомата PC4 има два варианта, съответно за двата вида трансферен протокол.



Фиг. 3.11.6. Схема на конвейерен автомат в общата точка

В схемите и на двата конвейерни автомата се използва синтезираната логическа схема на многоходов C-елемент.

### 3.12. Суперскаларен микроконвейер

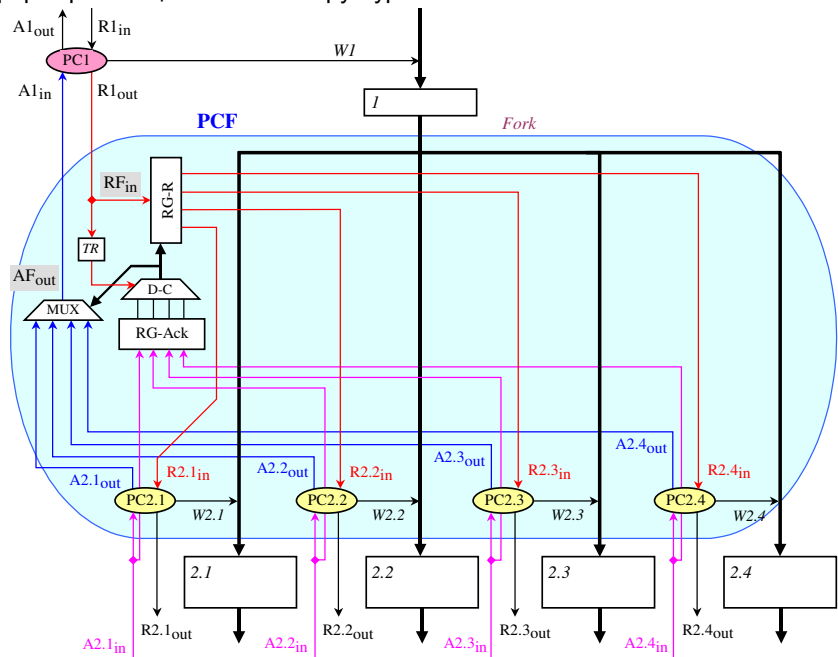
В този раздел се разглежда структура, съдържаща звено с неизбежно голяма латентност, в резултат на което пропускателната способност

на участъка, в който се намира това звено, силно се намалява. От това “страда” цялостният темп на функциониране на конвейера. Ако звеното не може да бъде декомпозирано и остава в своя вид, за облекчаване на пропускателната способност в този раздел е приложен подхода на апаратното насищане, известен под наименованието суперскаларност. Суперскаларността в случая се разбира като многократно добавяне на същото микроконвейерно звено в паралелни връзки.

Суперскаларната структура се характеризира с две особени точки, за управлението на които са необходими нови решения.

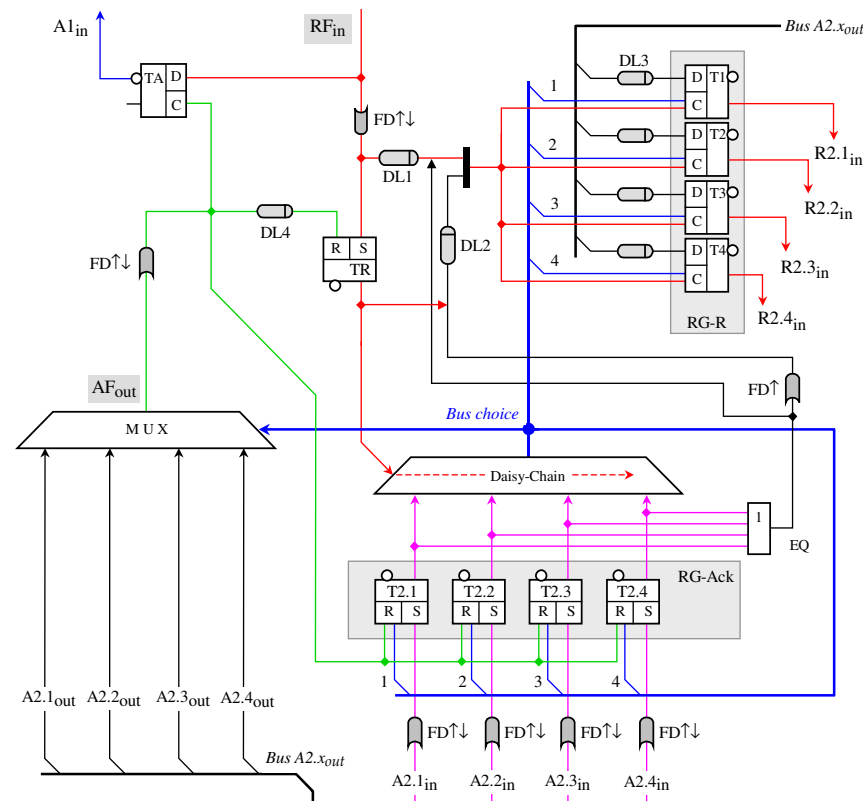
Показано е, че за управление на звеното в общата може да бъде използвано вече синтезирано решение, което е представено в раздел 3.4. Освен това в тази точка трябва да бъде приложено още едно решение, което се отнася за възстановяване на реда на получаваните резултати, тъй като е показано, че суперскаларната конвейерна структура не е в състояние да поддържа правилния ред. Микроконвейерният буфер, който възстановява реда на резултатите, е представен в раздел 3.5.

За управление на звената в точката на разклонение е синтезирана разпределена логическата структура на конвейерен автомат, представена на фигура 3.12.2. В разпределената част на автомата попадат автоматите на звената в отделните клонове, а останалите логически възли формират общата част на структурата.



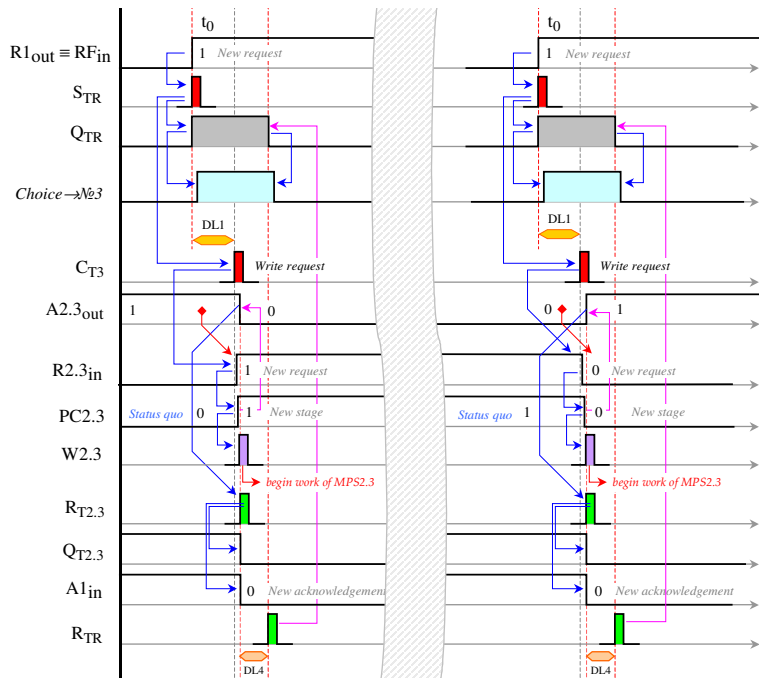
Фиг. 3.12.2. Структура на автомат в точката на разклонение

Синтезирани са логическите схеми на конвейерния автомат в два варианта, съответно за 2-фазов и за 4-фазов трансферен протокол, представени на фигури 3.12.4 и 3.12.6 съответно.

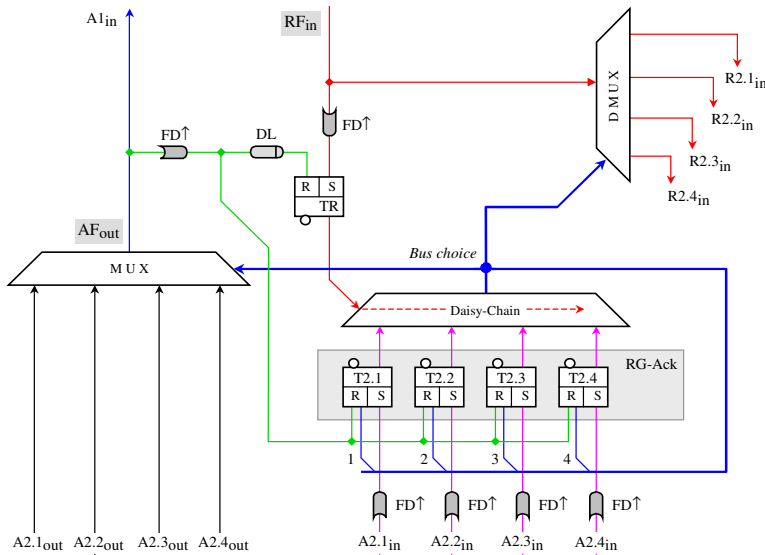


Фиг. 3.12.4. Логическа схема на общата част на 2-фазов конвейерен автомат в точката на разклонение

На фигура 3.12.5 е представена времедиаграмата за стартиране на микроконвейерно звено №2.3 от паралелната структура, чието потвърждение е било избрано чрез схемата за арбитражиране. Времедиаграмата е представена в два варианта, когато 2-фазовият конвейерен автомат на звено 2.3 е бил заварен в състояние нула и съответно в състояние единица. И в двата варианта началните условия за процеса на превключване са еднакви и се изразяват в предположението, че автоматът на предходното звено PC1 подава заявка със стойност единица. Суперскаларният конвейерен автомат е най-сложният от представените тук 2-фазови автомати.



Фиг. 3.12.5. Старт на звено 2.3



Фиг. 3.12.6. 4-фазов конвейерен автомат в точката на разклонение

## Глава 4

### Операционни схеми

В тази глава изследването има два аспекта. От една страна то е подчинено на концепцията за обединяване на операциите на линейни структури, а от друга страна се търси условие за тяхната асинхронна организация. Като имаме предвид изключителното значение на операция събиране като първична операция и като операция с една от най-големите латентности, оказваща влияние на всички останали аритметични операции, тук са разглеждани нейни реализации.

#### 4.1. Асинхронен двоичен комбинационен суматор

В този раздел е разглеждана хардуерната реализацията на модела на фактическото закъснение (*Complete Detection*). Тя е свързана със значителни апаратни разходи, ето защо е добре дошла всяка идея, способстваща тяхното минимизиране. Първоначално за модела на фактическото закъснение е синтезирана логическата функция за край на операцията, имаща вида

$$CD = F(p_{n-1} \cup \overline{p_{n-1}}) \cap \dots \cap F(p_1 \cup \overline{p_1}) \cap F(p_0 \cup \overline{p_0}). \quad (4.1.5)$$

където

$$p_{2i} = \overline{x_{2i} \cdot y_{2i} \cup x_{2i} \cdot p_{2i-1} \cup y_{2i} \cdot p_{2i-1}}, \quad (4.1.3)$$

$$p_{2i+1} = \overline{x_{2i+1} \cdot y_{2i+1} \cup x_{2i+1} \cdot p_{2i} \cup y_{2i+1} \cdot p_{2i}}. \quad (4.1.4)$$

Функцията  $CD$  реализира логическата конюнкция от поразрядните *dual-rail* кодове, изискващи паралелното изчисление на преносите (4.1.3) и (4.1.4), което значително увеличава апаратните разходи за тази схема. Този недостатък е отстранен като е използвана симетрична закъснителна верига. Постигнато е 50% минимизиране на апаратните разходи при запазване на бързодействието. Схемата е представена на фигура 4.1.6.

#### 4.2. Едновременно събиране на повече от 2 числа

В този раздел е представен синтеза и получените оценки за суматор, определен като многоразряден концентратор (3:1).

#### 4.3. Хоризонтален многовходов суматор

В този раздел е поставена задачата

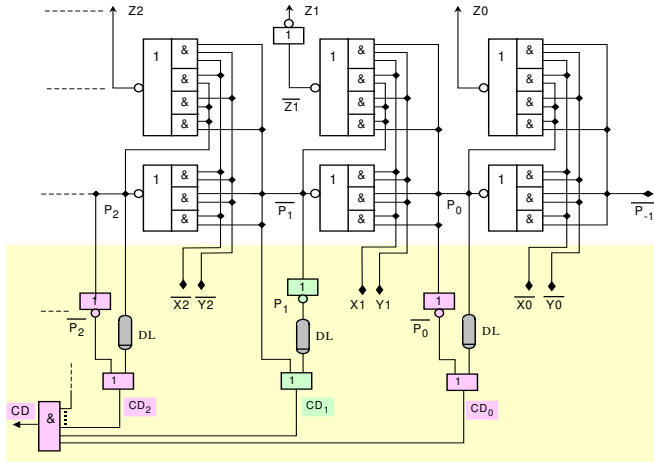
$$y = \sum_{i=1}^r x_i \quad (4.3.1)$$

където  $x_i, i = \overline{1, r}$  са  $n$ -битови числа от един и същ тип и формат.

Като алтернатива на последователното събиране със суматори е предложена пирамидална структура, изградена от концентратори (3:1). За нейния синтез и оценка е формулирана и доказана теорема, която

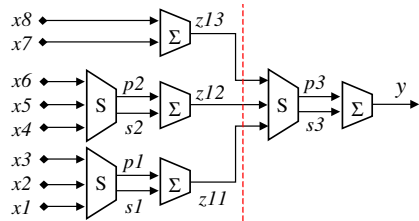
твърди, че максимално възможната дължина на сумата от  $r$  на брой  $n$ -разрядни цели числа е равна на

$$n + \lfloor \log_2(r) \rfloor + 1 \text{ [b]}. \quad (4.3.3)$$



Фиг. 4.1.6. Схема на асинхронен суматор

Осемвходова логическа структура е показана на фигура 4.3.4.



Фиг. 4.3.4. 8-входова сумираща схема

#### 4.4. Вертикален многовходов суматор

Задача (4.3.1) може да бъде изразена така

$$y = \sum_{j=0}^{n-1} \left( 2^j \cdot \sum_{i=1}^r b_j^{(i)} \right). \quad (4.4.4)$$

където с  $b_j^{(i)}$  е означен  $j$ -тият бит на  $i$ -тото събираемо  $x_i$ .

Вътрешната сума е наречена битова сума, а външната е сума от битовите суми. За получаване на битовите суми числата се подреждат едно под друго, което дава името на сумата "вертикална". На базата на концентратори (3:1) са предложени две структури, съответно за изчисляване на двете суми.

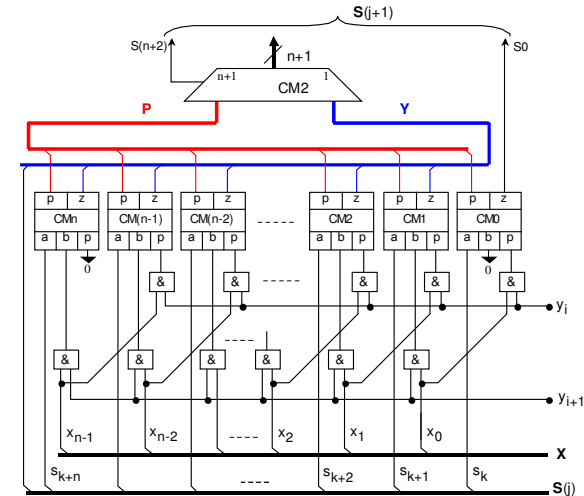
Както хоризонталния така и вертикалният суматор са определени като многовходови паралелни концентратори. И за двата типа са синтезирани алгоритми за оценка на апаратните разходи и за оценка на латентността. Изводите от сравнението на двата концентратора са представени в заключението.

#### 4.5. Конвейерен умножител с концентратори

Междинните суми на умножителя се изчисляват така

$$S_{j+1} = S_j + X_j \cdot y_{i+1} \cdot 2^{i+1} + X_j \cdot y_i \cdot 2^i. \quad (4.5.2)$$

За да може конвейерният умножител да използва концентратор от тип (3:1) за изчисляване на горните суми, е синтезирана логическа схема на концентратор, която отчита взаимното изместване на поразрядните суми. Схемата на конвейерния концентратор е показана на фигура 4.5.3.



Фиг. 4.5.3. Логическа схема конвейерния концентратор

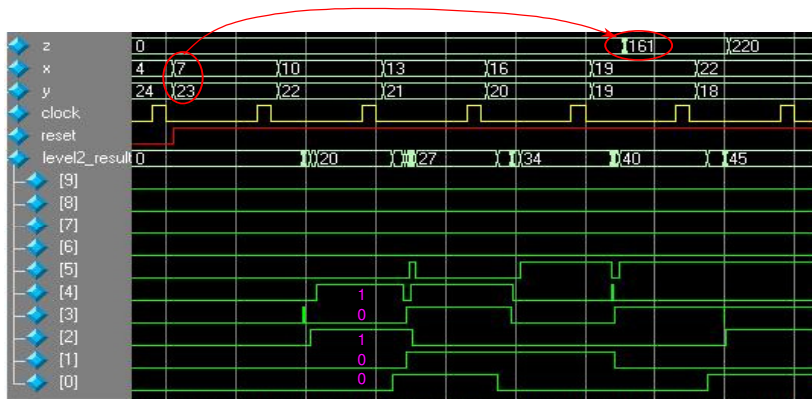
Синтезираният по този начин умножител реализира алгоритъм за умножение, който е от класа на методите за умножение с 2 разряда едновременно. В условия на асинхронен конвейер, умножителят е по-бърз от други представители на същия клас. Получена е аналитична оценка за допълнително спестеното време като функция от разрядността на операндите, която има вида  $\frac{3 \cdot n \cdot \tau}{2}$ .

Описаната структура е експериментирана за умножение на 8-битови числа. На фигура 4.5.5 е показан преходният процес в изходите на концентратора във второ ниво на конвейера. Така, при движение на двойките съмножители  $7.23=161$ ;  $10.22=220$ ;  $13.21=273$ ;  $16.20=320$ ;  $19.19=$



361;  $22.18=396$  и преминаването им през това ниво, на неговия изход последователно се получават междинните суми 20, 27, 34, 40 и 45.

Тъй като периодът на тактовата последователност в случая е 15[ns] и ясно се вижда, че е възможно да бъде намален двойно, то можем да твърдим, че такъв конвейер ще генерира 16-битово произведение на всеки 7,5[ns]. С други думи повече от  $133.10^6$  резултата в секунда.



Фиг. 4.5.5. Закъснения на второ ниво в конвейера

## Глава 5

### Средства за асинхронно управление

В тази глава са представени изследвания, посветени на класическите структури за логическо управление. Класическите подходи за управление чрез обособени (в смисъла на декомпозицията на В.М. Глушков) управляващи автомати не могат да бъдат отречени, защото те имат своите безспорни достойнства. Асинхронната идея на управление определя изложеното в тази глава.

#### 5.1. Генериране на сигнали за асинхронно управление

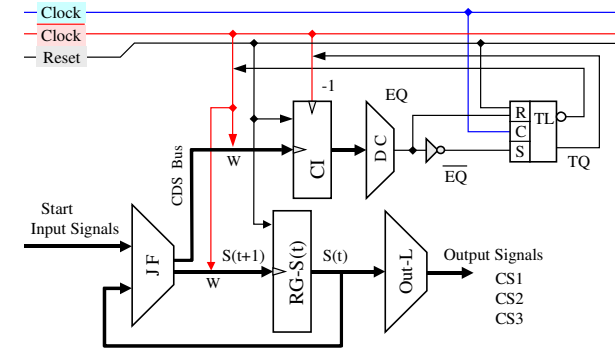
В този раздел е предложена една колекция от схеми за генериране на управляващи импулси, които са функция на събития в причинно-следствените връзки. Представените схеми са обобщени от гледна точка на логическите елементи, с които са реализирани. Представени са различни фронт-детекторни вериги, симетрични и асиметрични закъснителни вериги и техни комбинации, схеми за обозначаване на тригерно превключване, асинхронни селектори на импулси.

#### 5.2. Краен автомат с променлива продължителност на микротакта

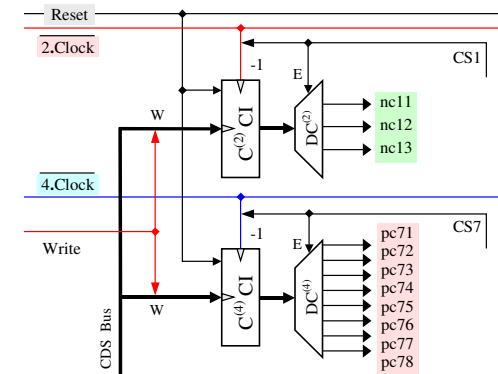
В този раздел е предложена една нова логическа структура на краен автомат с апаратно закрепена логика. Структурата е тактувана, но реали-

зира променлива продължителност на отделните микротактове и затова е определена като апериодична. Тя се характеризира с това, че съдържа D-Latch тригер, който отстранява състезанията в обратните връзки на управляващата логика и прави превключванията на автомата напълно синхронни с тактовите импулси. Структурата е апериодична до ниво пикокоманди.

На фигура 5.2.2 е показана основната схема, а на фигура 5.2.6, нейното допълнение.



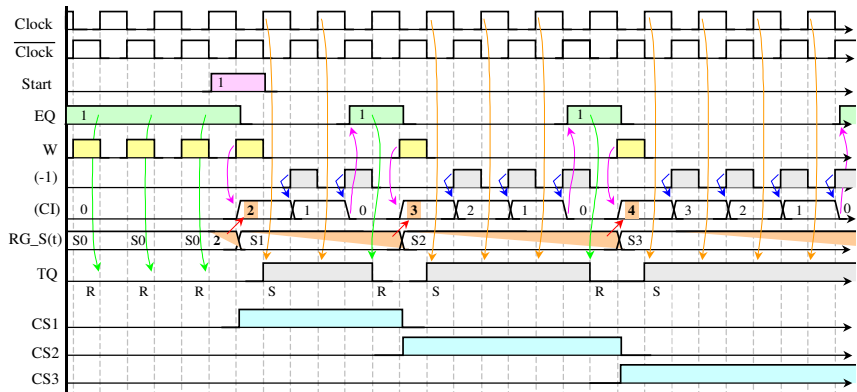
Фиг. 5.2.2. Логическа структура на автомата



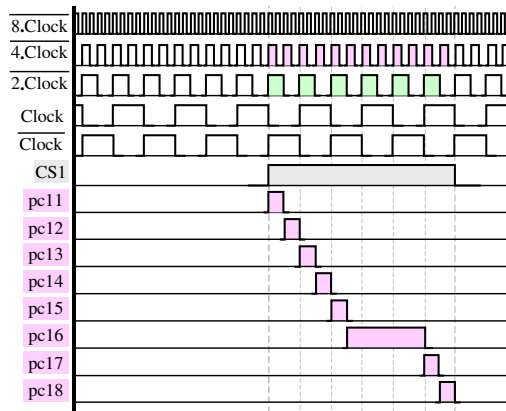
Фиг. 5.2.6. Структура за нано- и пикоуправляващи сигнали

На фигура 5.2.3 е представена времедиagramата на последователността (...S0<sup>(1)</sup>, S1<sup>(3)</sup>, S2<sup>(4)</sup>, S3<sup>(5)</sup> ...

Продължителността на микротактовете се шифрира като цяло число и е функция от старото състояние. Продължителностите на нано- и пикокомандите са кратни на честотите 2C и 4C и са функция на продължителността на микротакта, както е показано на фигура 5.2.5.



Фиг. 5.2.3. Времедиаграма на последователността (...S0<sup>(1)</sup>), S1<sup>(3)</sup>, S2<sup>(4)</sup>, S3<sup>(5)</sup>)

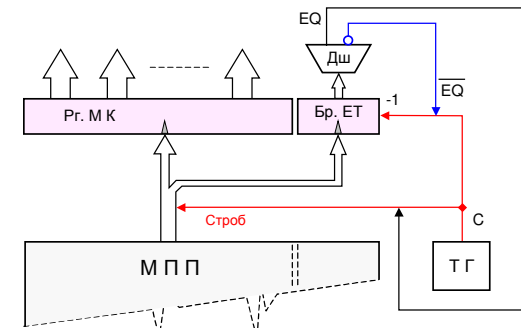


Фиг. 5.2.5. Осемтактова пикопрограма

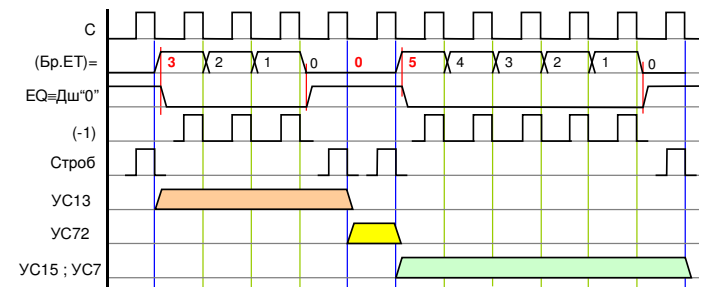
### 5.3. Микропрограмен автомат с променлива продължителност на микротакта

В този раздел е предложена нова структура на микрокомандата за автомат с програмно закрепена логика, която съдържа отделно поле за продължителността на микротакта. Предложени са две логически структури на микропрограмния автомат, които реализират аperiодичните превключвания по заден и по преден фронт на тактовите импулси. На фигура 5.3.2 е представена логическата структура, превключваща се по заден фронт, а на фигура 5.3.3, нейната времедиаграма. Подобен вид има и структурата, която съдържа логически възли, превключващи се по преден фронт.

Продължителността на микротакта е цяло число, което се определя от програмиста. То се намира в отделно поле в структурата на микрокомандата. В декрементния брояч на логическата структура това число се зарежда като начална стойност. На фигура 5.3.3 е показана времедиаграма с превключвания по заден фронт на тактовите импулси, а на фигура 5.3.5 превключванията са по преден фронт.



Фиг. 5.3.2. Структура с променлив микротакт по заден фронт



Фиг. 5.3.3. Микротактове с различна продължителност

### Основни резултати от научното изследване

Дисертационното изследване е посветено на формулиране и принципно теоретично решаване на фундаментални проблеми, свързани с хардуерната реализация на изчислителен процес с обща алгоритмична структура на микрооперационно ниво. То обединява в единна система проучвания, анализ, сравнителни качествени и количествени оценки, изводи, концепции, подходи, принципи и методи, форми на организация, алтернативни варианти, обоснован избор, формулиране и решаване на задачи за синтез. Прилагайки както известни, така и нови подходи, методи и средства, това изследване е фокусирано върху създаването на нови методи за хардуерна реализация на функционално пълна система

от алгоритмични структури и е ориентирано преди всичко към асинхронна и конвейерна организация на изчислителния процес. Получените от проведеното изследване резултати ни дават увереността, че формулираните за него научни задачи са решени и поставената научна цел е постигната.

Основните резултати на дисертационното изследване могат да бъдат групирани в следните насоки.

### Научни приноси

1. Формулирано и обосновано е ново научно направление, свързано със синтеза, анализа и рационализирането на микроконвейерни системи с обща структура. В това направление систематично са разработени основните елементи на нова методология за синтез и анализ на хардуерна реализация с неконвенционална организация на произволен, високо производителен изчислителен процес, която се систематизира за пръв път в научната литература. Същностна характеристика на създадените нови методи за синтез е постигането на гарантирано висока производителност. Тя е предопределена още от отсъствието на програмно управление на изчислителните структури и въвеждането на управление чрез обратна връзка, която се основава на асинхронно организиран конвейерен интерфейс, изграден върху принципа "ръкостискане". В основата на новите методи успешно е положен принципа за паралелна организация на изчисленията, преодолявайки възникващите проблеми при достигане на необходимата обобщеност. Методите реализират всички известни форми за паралелна организация на изчислителния процес като конвейерна, естествена и суперскаларна.

2. Формулирана и доказана е теорема за максимално възможната дължина на двоичната сума, получавана в общия случай от  $r$ -входови двоични концентратори на  $n$ -разрядни цели двоични числа, която е минималната реално необходима и достатъчна за представяне на резултата. Резултатът от теоремата е приложен при получаване на аналитичните оценки за апаратните разходи на синтезираните многовходови концентратори.

### Научно-приложни приноси

*В направление на микроконвейерните системи те могат да бъдат сведени до:*

1. Разработени са методи и са синтезирани следните оригинални конвейерни структури:

- 1.1. Линейни асинхронни микроконвейерни структури, произволно комбиниращи еднотактови и многотактови микроконвейерни звена;
- 1.2. Линейни асинхронни микроконвейерни структури от многотактови микроконвейерни звена, използващи нов конвейерен интерфейс, включващ статус-сигналите на звената;
- 1.3. Нелинейни асинхронни микроконвейерни структури, реализиращи неограничен брой условни алгоритмични преходи;

- 1.4. Микроконвейерен буфер за възстановяване на реда на изходните резултати;
- 1.5. Алгоритъм за управление на микроконвейерен буфер за възстановяване на реда на изходните резултати;
- 1.6. Паралелни асинхронни микроконвейерни структури, реализиращи естествен паралелизъм;
- 1.7. Суперскаларни асинхронни микроконвейерни структури.
- 1.6. Паралелни асинхронни микроконвейерни структури, реализиращи естествен паралелизъм;
- 1.7. Суперскаларни асинхронни микроконвейерни структури.

2. Разработени са методи и са синтезирани следните оригинални микроконвейерни звена:

- 2.1. Синхронно непрограмируемо многотактово звено за цикъл с известен брой повторения със следусловие, използващо и двата фронта на тактовия сигнал;
- 2.2. Синхронно програмируемо многотактово звено за цикъл с известен брой повторения със следусловие, използващо и двата фронта на тактовия сигнал;
- 2.3. Синхронно програмируемо многотактово звено за цикъл с известен брой повторения с предусловие, използващо и двата фронта на тактовия сигнал;
- 2.4. Синхронно непрограмируемо многотактово звено за цикъл с известен брой повторения със следусловие, използващо само един фронт на тактовия сигнал;
- 2.5. Синхронно програмируемо многотактово звено за цикъл с известен брой повторения със следусловие, използващо само един фронт на тактовия сигнал;
- 2.6. Синхронно програмируемо многотактово звено за цикъл с известен брой повторения със следусловие и с цифров компаратор, използващо само един фронт на тактовия сигнал;
- 2.7. Синхронно многотактово звено за цикъл с неизвестен брой повторения със следусловие, използващо и двата фронта на тактовия сигнал;
- 2.8. Синхронно непрограмируемо многотактово звено за цикъл с известен брой повторения със следусловие без вътрешни състезания, използващо и двата фронта на тактовия сигнал;
- 2.9. Синхронно програмируемо многотактово звено за цикъл с известен брой повторения със следусловие без вътрешни състезания, използващо и двата фронта на тактовия сигнал;
- 2.10. Синхронно многотактово звено за цикъл с неизвестен брой повторения с два регистъра със следусловие, използващо и двата фронта на тактовия сигнал.
- 2.11. Синхронно непрограмируемо звено с многотактово циклическо тяло за цикъл с известен брой повторения със следусловие, използващо и двата фронта на тактовия сигнал;
- 2.12. Синхронно звено с многотактово циклическо тяло за цикъл с неизвестен брой повторения със следусловие, използващо и двата фронта на тактовия сигнал;
- 2.13. Асинхронно звено с многотактово циклическо тяло за цикъл със следусловие, използващо двуфазов конвейерен протокол;
- 2.14. Асинхронно звено с многотактово циклическо тяло за цикъл със следусловие, използващо четирифазов конвейерен протокол;
- 2.15. Асинхронно звено с многотактово циклическо тяло за цикъл с предусловие, използващо двуфазов конвейерен протокол;
- 2.16. Асинхронно звено с многотактово циклическо тяло за цикъл с предусловие, използващо четирифазов конвейерен протокол.

3. Синтезирани са принципните логически схеми на следните асинхронни микроконвейерни автомати:

- 3.1. Четирифазов линеен конвейерен автомат с D-Latch тригер (фиг. 3.1.5);

- 3.2. Четирифазов линеен конвейерен автомат с D-Edge тригер (фиг. 3.1.7);
- 3.3. Четирифазов линеен конвейерен автомат с интерфейс за многотактови микро-конвейерни звена (фиг. 3.2.3);
- 3.4. Двухфазов нелинеен конвейерен автомата за звено с условен преход (фиг. 3.3.3);
- 3.5. Четирифазов нелинеен конвейерен автомата за звено с условен преход (фиг. 3.3.5);
- 3.6. Двухфазов нелинеен конвейерен автомата за звено в обща точка на конвейер с условни преходи (фиг. 3.4.4);
- 3.7. Асинхронен арбитър в двухфазов конвейерен автомат за обща точка в конвейер с условни преходи (фиг. 3.4.6);
- 3.8. Четирифазов ускорен линеен конвейерен автомат с D-Edge тригер (фиг. 3.4.10);
- 3.9. Четирифазов нелинеен конвейерен автомата за звено в обща точка на конвейер с условни преходи (фиг. 3.4.12);
- 3.10. Асинхронен арбитър в четирифазов конвейерен автомат за обща точка в конвейер с условни преходи (фиг. 3.4.6, но модифицирана само за преден фронт);
- 3.11. Двухфазов нелинеен конвейерен автомата за звено, генериращо условие за край на циклически повторения, в цикли със следусловие (фиг. 3.8.8);
- 3.12. Двухфазов нелинеен конвейерен автомата за звено, стоящо във входната точка на цикли със следусловие (фиг. 3.8.13);
- 3.13. Четирифазов нелинеен конвейерен автомата за звено, генериращо условие за край на циклически повторения, в цикли със следусловие (фиг. 3.8.14);
- 3.14. Четирифазов нелинеен конвейерен автомата за звено, стоящо във входната точка на цикли със следусловие (фиг. 3.8.16);
- 3.15. Двухфазов нелинеен конвейерен автомат за многотактово звено за цикъл с предусловие и с конвейерно тяло (фиг. 3.9.4);
- 3.16. Четирифазов нелинеен конвейерен автомат за многотактово звено за цикъл с предусловие и с конвейерно тяло (фиг. 3.9.7);
- 3.17. Двухфазов нелинеен конвейерен автомат за входната точка на паралелна конвейерна структура (фиг. 3.11.2);
- 3.18. Четирифазов нелинеен конвейерен автомат за входната точка на паралелна конвейерна структура (конкретизира се по фиг. 3.11.2);
- 3.19. Двухфазов нелинеен конвейерен автомат за общата точка на паралелна конвейерна структура (фиг. 3.11.6);
- 3.20. Четирифазов нелинеен конвейерен автомат за входната точка на паралелна конвейерна структура (конкретизира се по фиг. 3.11.6);
- 3.21. Двухфазов нелинеен конвейерен автомат за входната точка на суперскаларна конвейерна структура (фиг. 3.12.4);
- 3.22. Четирифазов нелинеен конвейерен автомат за входната точка на суперскаларна конвейерна структура (фиг. 3.12.6);
- 3.23. Двухфазов нелинеен конвейерен автомат за общата точка на суперскаларна конвейерна структура (аналогичен на фиг. 3.4.4);
- 3.24. Четирифазов нелинеен конвейерен автомат за общата точка на суперскаларна конвейерна структура (аналогичен на фиг. 3.4.12).

*В направление на микроконвейерните операционни схеми те могат да бъдат сведени до:*

**4.** Синтезирани са оригинални паралелни логически структури на многоходови концентратори. Разработени са алгоритми за изчисляване на количествените оценки на структурата им. Формулирани са правила за рационален избор на структура, които конструкторът може да спазва.

Синтезиран е конвейерен умножител с концентратори, естествено реализиращ алгоритъм от класа на методите за умножение с 2 разряда едновременно. Изведена е аналитична оценка за допълнителното му ускорение.

*В направление на класическите системи за управление те могат да бъдат сведени до:*

**5.** Обобщени са средствата за асинхронно управление чрез представянето им само с логически елементи, което позволява регулиране на техните параметри по време на синтеза в рамките на една автоматизирана технологична среда и изключва необходимостта от регулиране на ниво електронни елементи. Синтезирани са две оригинални принципни логически схеми за асинхронна селекция на импулси (фиг. 5.1.25 и фиг. 5.1.26). Предложен е метод за синтез на апериодичен краен автомат с апаратно закрепена логика без вътрешносхемни състезания в обратните връзки. За него е предложено структурно допълнение, позволяващо декомпозиция на микрокомандите до ниво апериодични пикокоманди, за което са предложени две оригинални структури (фиг. 5.2.2 и фиг. 5.2.6). Предложен е метод за синтез на апериодичен краен автомат с програмно закрепена логика. Логическите структури от фиг. 5.3.2 и фиг. 5.3.4 са оригинални.

*В направление на приложните системи те могат да бъдат сведени до:*

**6.** Технически реализации, използващи посочените по-горе оригинални логически структури и оригинални принципни логически схеми, са патентно защитими.

В хода на дисертационното изследване бяха разкрити неподозирани и непредвидени в началото проблеми. Голяма част от тях бяха преодоляни и техните решения се съдържат тук. По този начин дисертационното изследване предоставя решения за основните проблеми, повдигнати в него. Същевременно то оставя и недостатъчно изследвани някои страни на представените решения, за които са разкрити алтернативни варианти и допълнителни възможности, които са посочени в съответните раздели. Както всяко научно изследване, и настоящото може да бъде продължено, например в указаната насока.

## Публикации, свързани с дисертационния труд

1. **Тянев, Д.С.** (1979). *Схема за селекция на единичен синхронен импулс*. Електро-промишленост и приборостроене, Година XIV, №2, стр. 66-67, 1979.
2. **Тянев, Д.С.** (2004). *Организация на компютъра. Проектиране на логически структури*. ISBN: 954-20-0259-9. ТУ-Варна, 2004.
3. **Тянев, Д.С.** (2006). *Конвейерен умножител с концентратори*. Компютърни науки и технологии, ТУ-Варна, ISSN:1312-3335, Година IV, брой №1&2, стр. 23-28, 2006.
4. **Тянев, Д.С.** (2007). *Ново приложение на концентратори в схеми на конвейерни умножители*. International Conference TELECOM'07, St. St. Constantin & Elena, ISBN: 978-954-8329-93-4, Varna, BG, 2007.
5. **Тянев, Д.С.**, С.И. Колев, В. Йосифов (2007). *Метод за реализация на апаратни самоуправляващи се циклически структури*. Годишник на ТУ-Варна, Юбилеен сборник "45 години ТУ-Варна", ISSN 1311-896X, стр. 130-135, 2007.
6. **Тянев, Д.С.**, С.И. Колев, Д.В. Янев (2007). *Метод за реализация на апаратни самоуправляващи се циклически структури - част II*. Компютърни науки и технологии, ТУ-Варна, ISSN:1312-3335, Година V, брой №2, стр. 23-30, 2007.
7. **Тянев, Д.С.** (2007). *Организация на компютъра - упражнения*. ISBN: 954-20-0258-0, ТУ-Варна, 2007.
8. **Тянев, Д.С.** (2008). *Организация на компютъра*. Том 1: ISBN: 978-954-0412-7, Том 2: ISBN: 978-954-0413-4. ТУ-Варна, 2008.
9. Георгиев, С., **Д.С. Тянев**, В. Йосифов (2007). *Преглед на подходите и методите за проектиране на компютърни архитектури с ниска енергийна консумация*. Компютърни науки и технологии, ТУ-Варна, ISSN:1312-3335, Година V, брой №2, стр. 11-23, 2007.
10. Kolev, S.I., V. Josifov, **D. Tyanev** (2007). *Design of asynchronous pipelines for floating point multiplication and division*. Master thesis, FHTW-Berlin, Deutschland, 2007.
11. **Tyanev, D.S.**, V. Josifov, S.I. Kolev (2007). *Operational structures without controlling automata*. Proceedings of International Workshop on Network and GRID Infrastructures, Bulgarian Academy of Sciences, Sofia, Bulgaria, September 2007.
12. **Tyanev, D.S.**, S.I. Kolev (2008). *Principle scheme of aperiodic finite state machine*. Компютърни науки и технологии, ТУ-Варна, ISSN:1312-3335, Година V, брой №2, стр. 3-8, 2008.
13. **Tyanev, D.S.**, N.N. Nikolov, S.I. Popova (2008). *Synthesis and comparative analysis of multiple inputs parallel adders*. Proceedings of Fourth International Bulgarian-Greek Conference (CS'08), Kavala, Greece, Vol. 1, ISBN: 978-954-580-255-3, pp. 270-278, 2008.

14. **Tyanev, D.S.**, S.I. Kolev, D.V. Yanev (2009). *Micro-pipeline Section For Condition-Controlled Loop*. Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech'09), Ruse, Bulgaria, ISSN: 1313-8936, pp. 1.4-1(5), 2009.
15. **Tyanev, D.S.**, D.V. Yanev, S.I. Kolev (2009). *Method for realization of self-controlling loop apparatus structures*. Proceedings of the Fifth International Scientific Conference 2009 (CS'09), Sofia, Bulgaria, ISBN: 978-954-438-853-9, pp. 154-158, 2009.
16. **Тянев, Д.С.** (2009). *Четири-фазов микроконвейер с еднотактови и многотактови микроконвейерни звена*. Компютърни науки и технологии, ТУ-Варна, ISSN:1312-3335, Година VII, брой №1, стр. 3-12, 2009.
17. Procopov, S.P., **D.S. Tyanev** (2009). *Hardware implementation of strategies for servicing queues*. Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech'09), Ruse, Bulgaria, ISSN: 1313-8936, pp. 1.3-1(8), 2009.
18. **Тянев, Д.С.** (2009). *Управление на разклонения в асинхронни микроконвейери*. Компютърни науки и технологии, ТУ-Варна, ISSN:1312-3335, Година VII, брой №2, стр. 3-12, 2009.
19. Kolev, S.I., **D.S. Tyanev** (2010). *Early set to zero micropipelin*. Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech'10), Sofia, Bulgaria, ACM ISBN: 978-1-4503-0243-2, ACM Press, pp. 25-30, 2010.
20. **Tyanev, D.S.**, S.I. Popova (2010). *Asynchronous micro-pipeline with multi-stage sections*. Proceedings of the XLV International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST'2010), Ohrid, Macedonia, Vol. 2, ISBN: 978-9989-786-58-7, pp. 675-678, 2010.
21. **Tyanev, D.S.**, S.I. Kolev, D.V. Yanev (2010). *Race Condition free Asynchronous Micro-Pipeline Units*. Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech'10), Sofia, Bulgaria, ACM ISBN: 978-1-4503-0243-2, ACM Press, pp. 31-37, 2010.
22. **Тянев, Д.С.**, В.Т. Божикова (2010). *Алгоритъм за управление на микроконвейерен буфер*. Компютърни науки и технологии, ТУ-Варна, ISSN:1312-3335, Година VIII, брой №1, стр. 3-13, 2010.
23. **Tyanev, D.S.**, V.T. Bozhikova (2011). *Algorithm for micropipeline buffer control*. Applied Technologies and Innovations, Volume 4, April 2011, Prague Development Center, Czech Republic, ISSN: 1804-1191, pp. 12-21, 2011.
24. **Tyanev, D.S.** (2011). *Non-Linear Asynchronous Micro-Pipelines*. Наукові праці ДонНТУ, Донецьк, Україна, Серія "Інформатика, кібернетика та обчислювальна техніка", випуск 13(185), ISSN: 1996-1588, pp. 25-32, 2011.

25. **Tyanev, D.S.**, D.V. Yanev (2011). *Non-Linear Asynchronous Micro-Pipelines*, International Conference on Computer Systems and Technologies, (CompSysTech'11), Vienna, Austria, ACM ISBN: 978-1-4503-0917-2, pp. 38-44, 2011.
26. **Tyanev, D.S.**, S.I. Popova (2011). *Branch management into micro-pipeline joint dot*. Applied Technologies and Innovations, Volume 5, November 2011, Prague Development Center, Czech Republic, ISSN: 1804-1191, pp. 11-26, 2011.

### Научноизследователски проекти

- (2002). **Тянев, Д.С.** *Изследване на високоскоростни FPGA-матрици и разработване на методика за емулиране на логически модули в Xilinx ISE среда*. Научно изследователски проект №437/20.06. 2002 г., финансиран от ФНИ при ТУ-Варна.
- (2009). **Тянев, Д.С.** *Внедряване на интегрирани научни изследвания в обучението по компютърни науки*. Научно изследователски проект №НП21, финансиран от ФНИ при ТУ-Варна. Годишник на ТУ-Варна, Том 2, ISBN: 1311-896X, стр. 41-42, 2009.
- (2010). Колев, Й., А. Майналовски, **Д. Тянев**, И. Булиев, П. Попов, Д. Ковачев, Н. Николов, Н. Кенаров, А. Карачомаков, Х. Русев, Д. Иванов, Р. Раденков, И. Ставрев, Я. Стоянов. *Факултетна лаборатория за проектиране и тестване на специализирани интегрални схеми (ASICLAB)*. Научно изследователски проект ИСП№02, финансиран от ФНИ при ТУ-Варна, Годишник на ТУ-Варна 2010, Том 3, ISBN: 1311-896X, стр. 89-90, 2010.
- (2010). Рускова, Н.С., **Д.С. Тянев**. *Интердисциплинарни научни изследвания и внедряване в университетски учебни курсове*. Научно изследователски проект №НП10, финансиран от ФНИ при ТУ-Варна, Годишник на ТУ-Варна 2010, Том 3, ISBN: 1311-896X, стр. 29-30, 2010.
- (2011). Рускова, Н.С., **Д.С. Тянев**. *Научни изследвания в областта на компютърните науки с приложения в обучението*. Научно изследователски проект №НП9, финансиран от ФНИ при Технически Университет – Варна, 2011. Междинен отчет.

### Annotation

Tyanev, D. S. **Synthesis of asynchronous micro-pipeline systems with common structure**. Dissertation for the award of “Doctor of Science” degree. Faculty of Computing and Automation, Technical University of Varna, Varna, 2011, (ii+306) pages.

In dissertation there is unified methodological approach created composed of systematically developed methods for synthesis of hardware implemented random structure algorithms in form of asynchronous micro-pipeline systems. Along with these methods there are original operating schemes synthesized, as well as multi-input concentrators and their applications.

New scientific tendency is highlighted, relevant to synthesis, analysis and optimization of common structure micro-pipeline systems. There are original synchronous and asynchronous one- and multi-cycle micro-pipeline stages synthesized performing under different conditions; as well as logical schemes of pipeline controllers, combining one-cycle and multi-cycle micro-pipeline stages control.

There are formulated and solved new scientific tasks for hardware synthesis of asynchronous non-linear, cyclic, parallel and superscalar micro-pipeline structures, as well as their combinations. Original logical schemes of pipeline controllers are synthesized allowing structures' creation. The task for order restore of pipeline outgoing results is solved by synthesis of original logical structure of micro-pipeline buffer and its managing algorithm. Also is solved the task for request arbitration into micro-pipeline systems' joint-dot by asynchronous arbiter synthesis.

Method for synthesis of a-periodic finite state machine with hardware and software attached logic is developed.

Efficiency of synthesized micro-pipeline systems is investigated and validated by functional time-diagrams.

Obtained into dissertation results are methodological base for synthesis of asynchronous micro-pipeline systems with common structure of the computations, which is systematized for the first time in scientific literature.

**Keywords:** asynchronous micro-pipelines, non-linear micro-pipelines, cyclic micro-pipelines, parallel micro-pipelines, superscalar micro-pipelines, pipeline controllers, NRZ and RZ transfer protocols, micro-pipeline restore order buffer, delay models, concentrators, a-periodic finite state machines.

© Dimitar Tyanev.

© Technical University of Varna, 2011.

## Аннотация

Тянев, Д. С. **Синтез асинхронных микроконвейерных систем с общей структурой.** Диссертация на соискание ученой степени “Доктор наук”. Факультет вычислительной техники и автоматизации, Технический университет - Варна, Варна, 2011, (ii+306) стр.

В диссертации предлагается единый методологический подход, составленный из систематически разработанных методов синтеза аппаратной реализации алгоритмов произвольной структуры в виде асинхронных микроконвейерных систем. С помощью этих методов синтезируются также оригинальные операционные схемы, многовходовые концентраторы и их приложения.

Выделено новое научное направление, связанное с синтезом, анализом и оптимизированием микроконвейерных систем с общей структурой. Синтезированы оригинальные синхронные и асинхронные одноктактные и многотактные микроконвейерные звенья, функционирующие в разнообразных условиях, логические схемы конвейерных автоматов, комбинирующих управление различными звеньями.

Сформулированы и решены новые научные задачи синтеза асинхронных линейных, нелинейных, циклических, параллельных и суперскалярных микро-конвейерных структур, а также их комбинаций. Синтезированы оригинальные логические схемы конвейерных автоматов, позволяющие построение таких структур. Решена задача восстановления порядка сходящихся с конвейера результатов путем синтеза оригинальной логической структуры микроконвейерного буфера и создания алгоритма для его управления. Через синтез асинхронного арбитра решена задача справедливого арбитражного заявления заявок на обслуживание в общих точках микроконвейерных систем.

Разработан метод синтеза аperiodических конечных автоматов с аппаратно и с программно закрепленной логикой. Работоспособность синтезированных микроконвейерных систем исследована и обоснована с помощью функциональных представлений в виде временных диаграмм.

Полученные в диссертации результаты представляют собой методологическую основу для синтеза асинхронных микроконвейерных систем с общей структурой вычислительного процесса, которая впервые систематизируется в научной литературе.

**Ключевые слова:** асинхронные микроконвейеры, нелинейные микроконвейеры, циклические микроконвейеры, параллельные микроконвейеры, суперскалярные микроконвейеры, конвейерные автоматы, NRZ и RZ протоколы трансферта, микроконвейерный буфер восстановления порядка, модели задержки, концентраторы, аperiodические конечные автоматы.

© Димитр Тянев.

© Технический университет - Варна, 2011.

## Анотация

Тянев, Д. С. **Синтез на асинхронни микроконвейерни системи с обща структура.** Диссертация за присъждане на научната степен “Доктор на науките”. Факултет по изчислителна техника и автоматизация, Технически университет - Варна, Варна, 2011, (ii+306) стр.

В диссертацията е създаден единен методологичен подход, съставен от систематично разработани методи за логически синтез на хардуерната реализация на алгоритми с произволна структура във вид на асинхронни микроконвейерни системи. По този начин е открито ново научно направление, свързано със синтеза, анализа и рационализирането на микроконвейерни системи с обща структура. Синтезирани са оригинални синхронни и асинхронни едноктактови и многотактови микроконвейерни звена, функциониращи в разнообразни условия, логически схеми на конвейерни автомати, комбиниращи управлението на различни звена. Синтезирани още оригинални операционни схеми, многовходови концентратори и техни приложения.

Формулирани и решени са нови научни задачи за синтез на асинхронни линейни, нелинейни, циклически, паралелни и суперскалярни микроконвейерни структури, както и техни комбинации. Синтезирани са оригинални логически схеми на конвейерни автомати, позволяващи изграждането на тези структури. Решена е задачата за възстановяване на реда на слизащи от конвейера резултати чрез синтез на логическа структура на микроконвейерен буфер и алгоритъм за неговото управление. Решена е задачата за справедливо арбитражиране на заявките за обслужване в общи точки на микроконвейерни системи чрез синтез на асинхронен арбитър.

Разработен е метод за синтез на аperiodични крайни автомати с аппаратно и с програмно закрепена логика.

Работоспособността на синтезираните микроконвейерни системи е изследвана и обоснована чрез времедиаграмни функционални представления.

Получените резултати в диссертацията са методологична основа за синтез на асинхронни микроконвейерни системи с обща структура на микрооперационен изчислителен процес, която се систематизира за пръв път в научната литература.

**Ключови думи:** асинхронни микроконвейери, нелинейни микроконвейери, циклически микроконвейери, паралелни микроконвейери, суперскалярни микроконвейери, конвейерни автомати, NRZ и RZ трансферни протоколи, микроконвейерен буфер за възстановяване на реда, закъснителни модели, концентратори, аperiodични крайни автомати.

© Димитър Тянев.

© Технически университет - Варна, 2011.