

# МЕТОД ЗА РЕАЛИЗАЦИЯ НА АПАРАТНИ САМОУПРАВЛЯВАЩИ СЕ ЦИКЛИЧЕСКИ СТРУКТУРИ - Част II

Димитър Тянев, Стамен Колев, Драгомир Янев

**Резюме:** Изследването представя нов метод за проектиране на операционни (изчислителни) структури, реализиращи определени разновидности на циклически алгоритми от вида с предварително известен брой повторения. Съществената характеристика на получените операционни структури се изразява в това, че те функционират без управляващ краен автомат. Методът може да бъде прилаган при проектиране както на синхронно работещи, така и на асинхронно работещи операционни структури, т.е. това могат да бъдат тактувани или нетактувани структури. Основното достоинство на метода се състои в опростяване и хомогенизиране на апаратурата, реализираща изчислителния процес. Получените структури са високо производителни, тъй като могат да бъдат конвейерно организирани.

## METHOD FOR REALIZATION OF SELF-CONTROLLING LOOP APPARATUS STRUCTURES – PART II

Dimitar Tyanev, Stamen Kolev, Dragomir Yanev

**Abstract:** The present research represents a novel method for operational structures design, which implement a versatile set of count-controlled loop algorithms. Main feature of these structures is that they operate without controlling automata. The method can be used for the design of synchronous and asynchronous structures. It simplifies the structures, gives us the ability to organize the structures in pipeline fashion, which results in better throughput.

### 1. Увод

Основната идея, за редуция на управляващите крайни автомати (като съставна част от всяко цифрово устройство), която авторите изследват, е изложена и обсъждана от тях в [1], а отделни нейни разновидности в [2] и в [3]. Основната цел при изследването на тази идея е да покаже, че известната декомпозиция на Глушков [4], на която може да бъде подложен всеки изчислителен автомат, е възможно да бъде редуцирана и опростена за сметка на неговата управляваща част. В резултат на това логическата структура на един автомат ще бъде сведена само до своята модифицирана операционна част ([1] фигура 3, както и [4] пункт 3.8.3), првърщайки я, според разбирането на авторите, в самоуправляваща се. Постигането на тази цел е възможно, ако бъдат разработени методи за проектиране на безавтоматно управление на хардуерните реализации на основните алгоритмични структури, срещащи се в изчислителния процес. Към настоящият момент такива са предложени за реализация на линейни и разклонени алгоритмични структури [1], както и за някои видове циклически структури [2]. Поради своето многообразие последните продължават да бъдат обект и в настоящото изследване.

### 2. Изложение

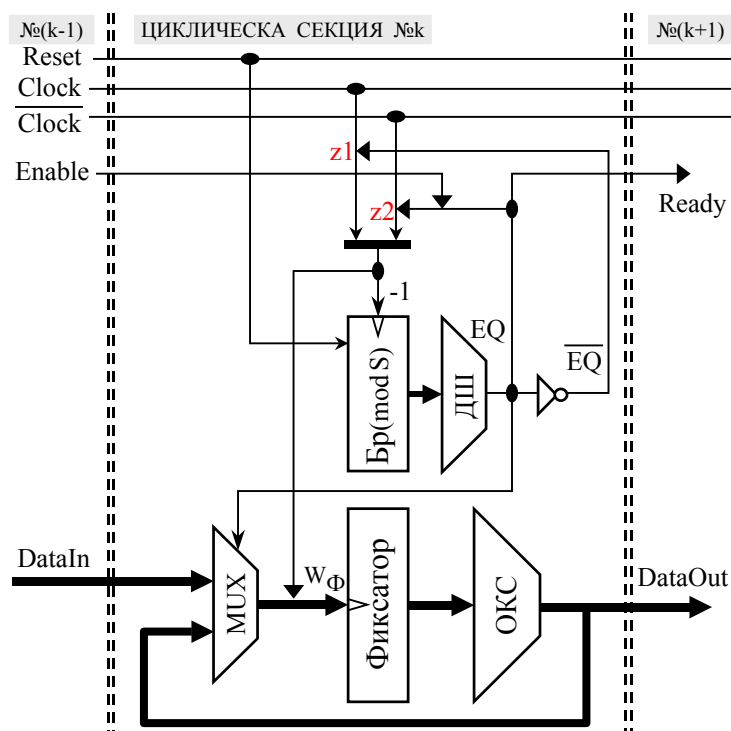
Предложените в [2] решения за циклически алгоритмични структури от вида с предварително известен брой повторения използват в своите схеми декрементни броячи, които се превключват по задния фронт на входните импулси (-1). Тъй като в много синхронни елементи, а така също и в програмируемите логически клетки на CPLD и FPGA елементите, се реализират превключвания по предния фронт на тактуващите импулси, тук са предложени съответните алтернативни решения. Това са логически структури, в които всички структурни възли се превключват по преден фронт.

Въпросът е актуален с това, че при използване на еднофазов тактов генератор за синхронизация на превключванията в логическите структури, маркирането на отделни фиксирани моменти в рамките на тактовия период е затруднено, а то е крайно необходимо в

смисъла на основната цел на това изследване. Тук се има предвид това, че в самоуправляващите се структури се налага да се изграждат различни обратни връзки, които създават условия за възникване на състезаващи се превключвания. Последните не винаги е възможно да бъдат овладяни, така че синхронизацията изисква приложение на различни подходи, включващи използването на противосъстезателни елементи, комбинирано използване на схеми, превключващи се по различни фронтове, както и използването на няколко фазов тактов генератор.

## 2.1 Самоуправляваща се логическа структура за реализация на цикли с предварително известен брой повторения

Методът за реализация на самоуправляваща се циклическа структура с линейно тяло е изложен в [2]. Като решение на изявения проблем със синхронизацията и на възможното многообразие тук са представени синтезираните и изследвани две логически структури: за цикли с конструктивно предопределен брой повторения и за цикли с променлив брой на повторенията. На фигура 1 е представена логическа структура от първата разновидност. Структурата изпълнява действията в тялото на цикъла чрез аппаратната им реализация в комбинационната схема ОКС. Входните данни за всяка поредна итерация се поддържат от регистъра фиксатор, който записва новите резултати по предния фронт на селектираните импулси  $W_{\Phi}$ . Условието за край на цикъла се реализира, модифицира и контролира от декрементния брояч и комбинационните схеми, свързани с него. В изходно състояние съдържанието на брояча Бр е нула,  $EQ=1$  и входният мултиплексор е включен към шината *DataIn*. Въпреки че структурата се тактува непрекъснато, в нея не се извършват превключвания. Изчисленията започват с появата на разрешаващия сигнал *Enable*, както е показано на фигура 2. За правилно функциониране сигналът *Enable* следва да се появява по предния фронт на тактовите импулси и да има продължителност не повече от един период.



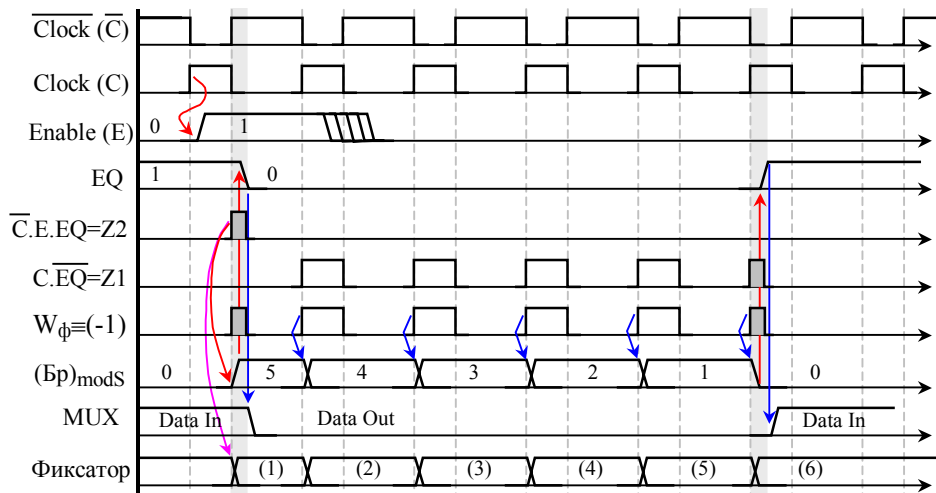
Фиг. 1 Циклическа самоуправляваща се операционна структура, работеща по преден фронт

Както се вижда от примерната времедиаграма, за надеждна синхронизация на превключванията е използван и инверсният тактов сигнал. В изразения процес на превключване има два особени момента, които следва да бъдат пояснени.

Първият се отнася до скъсената продължителност на първия и последния декрементни импулси (вижте абсцисата на сигнал (-1)). Тъй като в структурата от фигура 1 броячът се превключва по предния фронт на входните импулси, следствието - разпространяващият се

по обратната връзка сигнал EQ (респективно и неговата инверсна стойност), затваря чрез нулевата си стойност със закъснение съответната конюнкция (вижте точка z1 и точка z2).

$$(-1) = z1 \cup z2 = (\text{Clock} \cap \overline{\text{EQ}}) \cup (\overline{\text{Clock}} \cap \text{Enable} \cap \text{EQ}) . \quad (2.1.1)$$



Фиг. 2 Времедиаграма за изпълнение на цикъл от 6 повторения

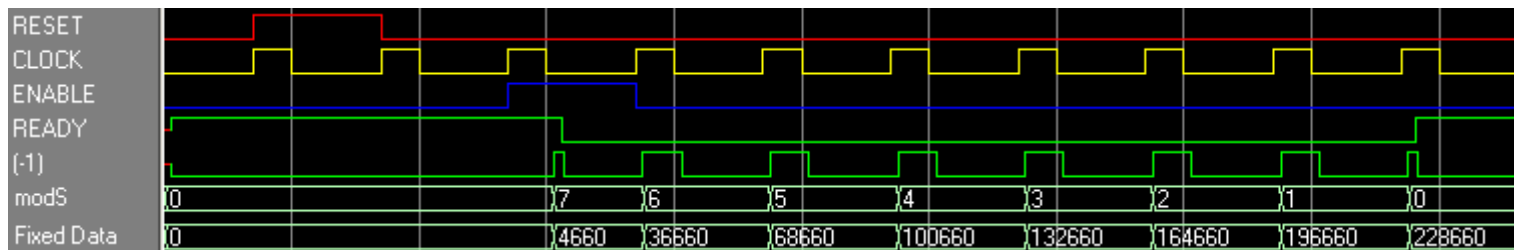
В резултат на това, продължителността на тези скъсени импулси, е равна на времето за превключване на брояча, плюс времето за превключване на дешифратора, плюс времето за превключване на останалите логически елементи в обратната връзка. Това време е напълно достатъчно и въпреки че обратната връзка поражда това състезание, то не се самовъзпроизвежда, благодарение на използваните в брояча противосъстезателни тригери със структура *Edge*.

Вторият особен момент се отнася до продължителността на инверсната фаза на тактовия сигнал. Тъй като тя определя престоя на началните входни данни във фиксатора (вижте абсцисата на фиксатора), то времето за първото операционно изчисление в схемата ОКС ще бъде по-малко от продължителността на периода. От тук следва, че за получаване на правилни резултати, трябва да се предяви изискване както към тактовата честота, така и към коефициента на запълване на тактовия сигнал, който следва да е възможно най-малък. За всяко следващо изчисление (циклично повторение), освен за първото, ще бъде отделено време, равно на продължителността на цял период.

Практическата проверка на синтезираната структура е изпълнена в средата ISE 10.1 на фирма *Xilinx* за интегрални схеми от серията *Spartan 3E*. На фигура 3 по-долу е представена работата на генерирания модел. Изпълнен е цикъл с 8 повторения. За по-лесна проверка в операционната комбинационна схема ОКС е заложено итерационното изчисление на натрупваща се сума:

$$S_k = S_{k-1} + 32000 \quad , \quad \text{при } S_0 = 4660$$

Така върху последната абсциса *Fixed Data* от фигура 3 се виждат получаваните числени резултати: 4660, 36660, 68660 и т.н.

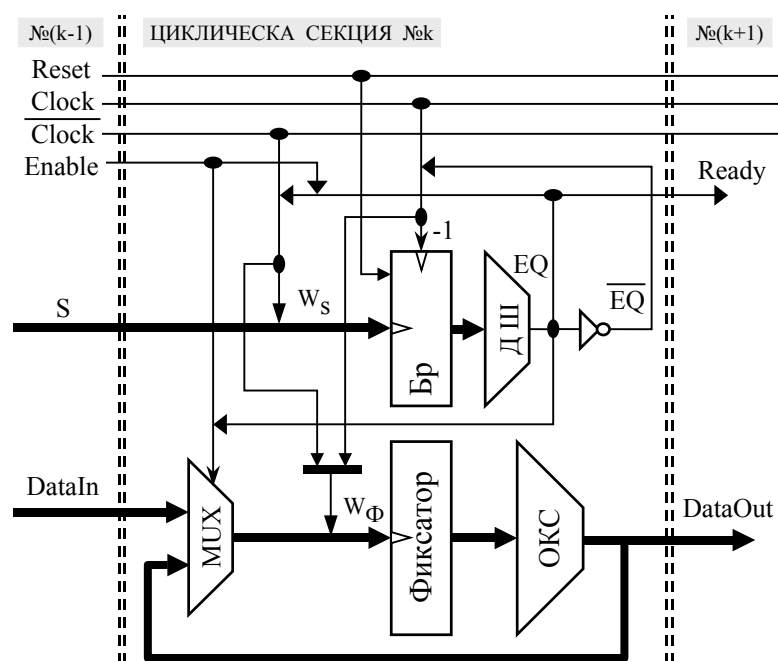


Фиг. 3 Времедиаграма на *Xilinx*-модела

## 2.2 Самоуправляваща се логическа структура за реализация на цикли с променлив брой повторения

Тук на фигура 4 по-долу е представена синтезираната и изследвана логическа структура, съответстваща на циклическа алгоритмична структура от вида с предварително известен брой повторения, който обаче е променлив и е функция на предходни етапи на изчислителния процес. Циклическата организация с променлив брой повторения изисква използването на декрементен програмируем брояч, в който определената стойност  $S$  се записва в качеството си на негова начална стойност. Максимално възможната стойност е ограничена от модула за броене. Отчитайки споменатата програмируемост, логическата структура има отделен вход към брояча Бр. Освен това всички нейни логически възли с памет се превключват по преден фронт на синхро сигналите. Условието за край на цикъла се формира от дешифратора ДШ, който разпознава съдържание нула и формира сигнала EQ. Управлението на събитията, свързани със съдържанието на брояча, се постига чрез логическите схеми в неговите обратни връзки.

В изходно състояние съдържанието на брояча Бр е нула,  $EQ=1$  и входният мултиплексор е включен към шината *DataIn*. Въпреки че структурата се тактува непрекъснато, в нея не се извършват превключвания. Изчисленията започват с появата на разрешаващия сигнал *Enable*, както е показано на фигура 2. За правилно функциониране сигналът *Enable* следва да се появява по предния фронт на тактовите импулси и да има продължителност не повече от един период.



Фиг. 4 Програмируема циклическа самоуправляваща се операционна структура, работеща по преден фронт

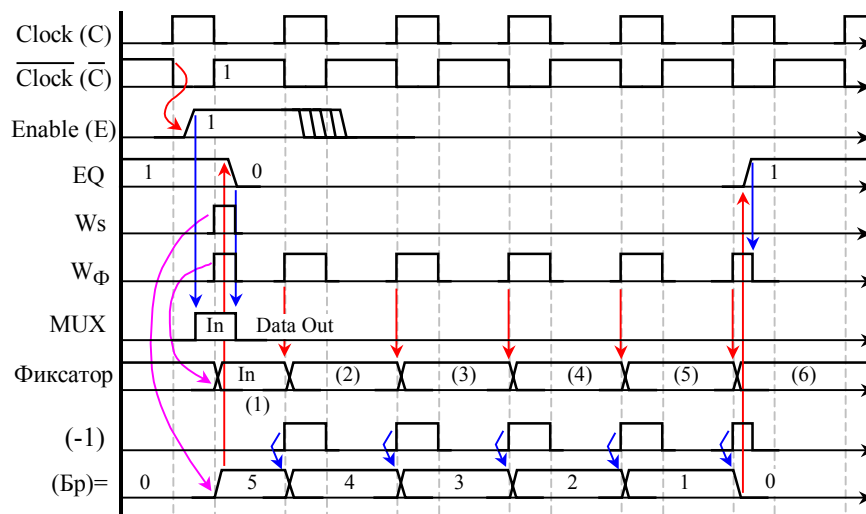
На фигура 5 е представена примерна времедиаграма за изпълнение на цикъл от 6 повторения. Както може да се види, условието за запис на нови данни и старт на изчисленията в структурата, се дава от сигналите  $W_s$  и  $W_\phi$ , които се формират според следната логика:

$$W_s = \overline{\text{Clock}} \cap (\text{Enable} \cap \text{EQ}) . \quad (2.2.1)$$

$$W_\phi = W_s \cup (-1) = (\text{Clock} \cap \overline{\text{EQ}}) \cup (\overline{\text{Clock}} \cap \text{Enable} \cap \text{EQ}) . \quad (2.2.2)$$

Използването на сигнала EQ в разглежданата структура е аналогично на изложеното за предишната. За тази структура цикълът с предусловие е възможен. В този случай, възможността за неизпълнение на действията от тялото на цикъла налагат операционната

схема ОКС да бъде преместена в обратната връзка, а на изход *DataOut* да излиза съдържанието на регистъра-фиксатор. Така, ако записаната начална стойност в брояча се окаже равна на нула ( $S=0$ ), то записаните във фиксатора входни данни ще излязат без обработка на изход.

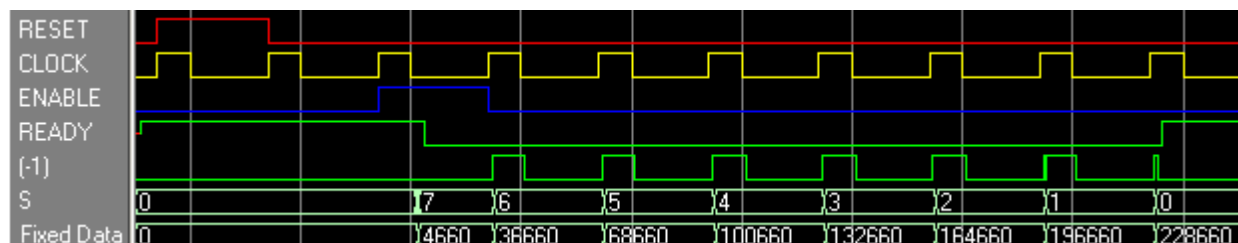


Фиг. 5 Времедиаграма на структура, работеща по преден фронт

Практическата проверка на синтезираната тук структура е изпълнена в средата ISE 10.1 на фирма *Xilinx* за интегрални схеми от серията *Spartan 3E*. На фигура 6 по-долу е представена работата на генерирания модел. Изпълнен е цикъл с 8 повторения при начална стойност  $S=7$ . Както в предишния модел, за по-лесна проверка, в операционната комбинационна схема ОКС е заложено аналогичното итерационно изчисление на натрупващата се сума:

$$S_k = S_{k-1} + 32000 \quad , \quad \text{при } S_0 = 4660$$

Така върху последната абсциса от фигура 6 се виждат получаваните числени резултати.



Фиг. 6 Времедиаграма на *Xilinx*-модела

### 2.3 Самоуправляваща се логическа структура за реализация на цикли с променлив брой повторения на базата на цифров компаратор

На фигура 7 е показан още един възможен вариант за структурата, реализираща цикли с предварително известен, но променлив брой повторения. Това е структура, в която е използван инкрементен брояч. Неговото изходно съдържание, както и това на регистъра на контролната стойност  $PtS$ , е винаги нула, което се постига чрез импулсите на сигнала *CLR* (*Clear up*). След стартиране на процеса от сигнала *Enable*, входните данни *DataIn* се записват във фиксатора, контролната стойност  $S$  се записва в регистър  $PtS$ , а съдържанието на брояча започва да расте. Последното непрекъснато се сравнява с контролната стойност  $S$  от схемата за сравнение *CC*, която генерира в края на процеса сигнала за съвпадение *Agree*. Изчислителният процес протича под управлението вътрешни и външни сигнали, чиято логика е следната:

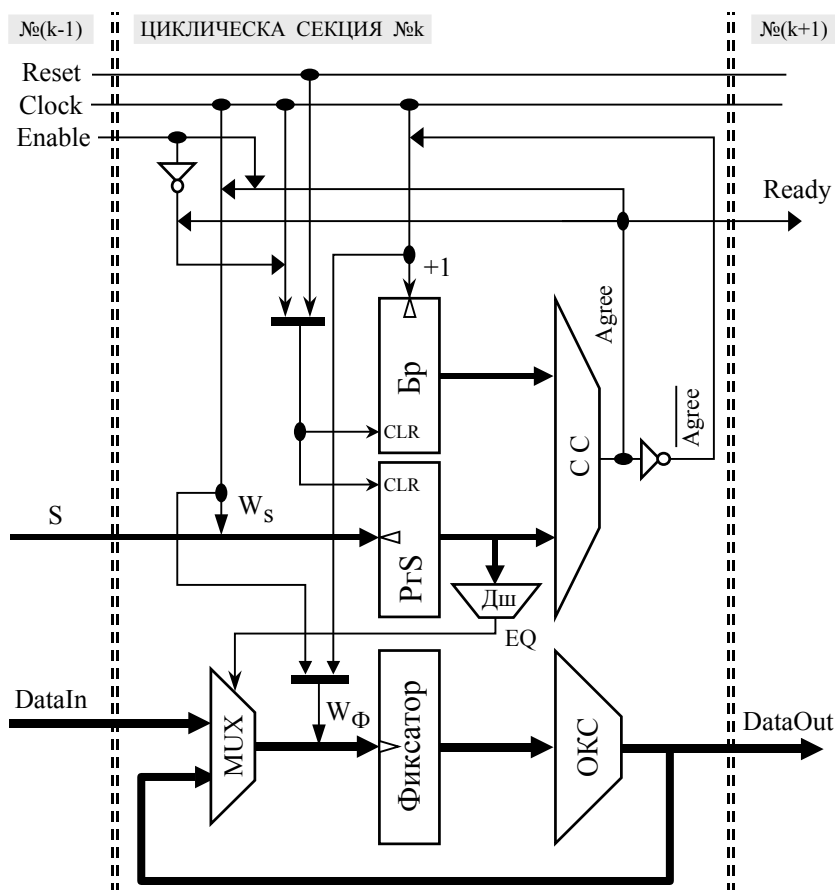
$$CLR = Reset \cup [C \cap (\overline{Agree} \cap \overline{Enable})] \quad (2.3.1)$$

$$(+1) = C \cap \overline{\text{Agree}} \quad (2.3.2)$$

$$W_s = C \cap \text{Agree} \cap \text{Enable} \quad (2.3.3)$$

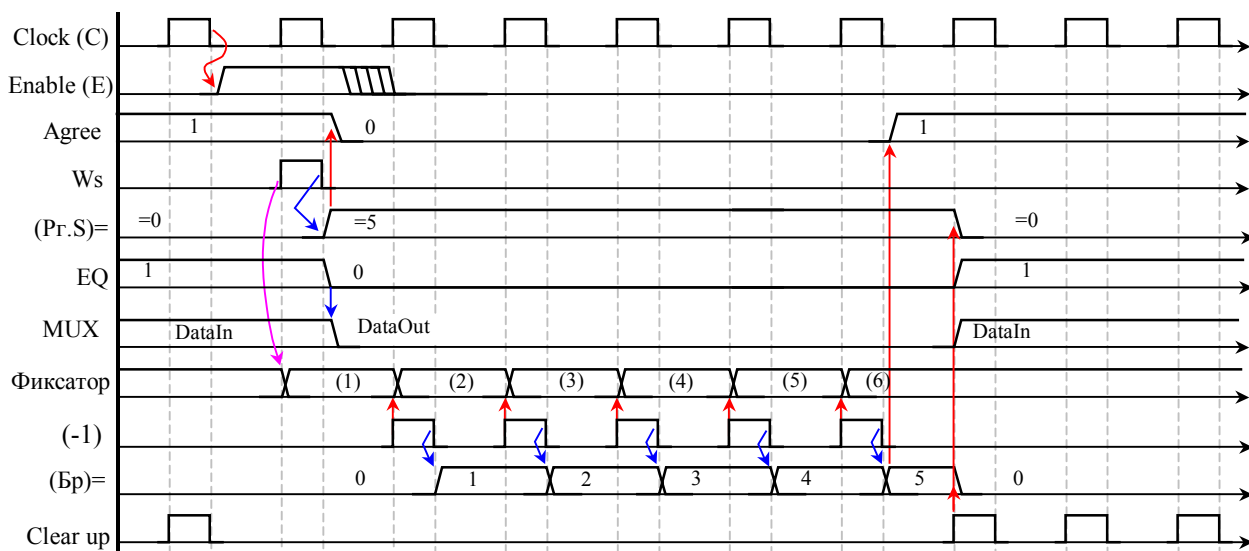
$$W_\phi = W_s \cup (+1) \quad (2.3.4)$$

Надеждността на превключванията и избягването на състезанията в структурата са осигурени чрез логически възли с различно превключване. Регистърът PrS и броячът Бр се превключват по заден фронт, а регистърът на фиксатора – по преден.



Фиг. 7 Циклическа операциона структура с променлив брой на повторенията

Процесът на функциониране на синтезираната структура е представен чрез времедиagramата от фигура 8.



Фиг. 8 Времедиagramа за изпълнение на цикъл с 6 повторения

### 3. Заключение

От сравнението на всички резултати, публикувани по-рано и получените тук, отнасящи се до циклическите структури, се налага следния извод: структури, които използват логически възли с памет, превключващи се по един и същ фронт на управляващите сигнали, са склонни да генерират вътрешни за себе си състезания. В същото време, структурите, които комбинират превключващите фронтове не генерират състезания във вътрешността си. Възможните за някои структури състезания обаче не се самовъзпроизвеждат благодарение на използваните противосъстезателни тригерни елементи. В този смисъл за такива структури можем да говорим, че притежават частична асинхронност.

Представените тук резултати доказват възможността за апаратна реализация на самоуправляващи се изчислителни структури, свободни от автоматно управление. С това се засилва увереността на авторите, че апаратната реализация на произволни алгоритмични структури като самоуправляващи се, е възможна. Тук следва да се заяви, че все още съществуват алгоритмични структури, чието изследване в смисъла на основната идея е предстоящо.

### Литература

- [1]. Josifov V., Kolev St., Tyanev D., *Operational structures without controlling automata*, International workshop on network and grid infrastructures, Institute for parallel processing at the bulgarian academy of sciences department “distributed computing systems and networks”, Sofia, Bulgaria, 27-28 Sept. 2007, EC Project BIS21++ “Bulgarian IST centre of competence in 21 century“, <http://bis-21pp.acad.bg/events/events.htm> ; <http://www.tyanev.com/home.php?lang=en&mid=23&mod=0&sub=4> .
- [2]. Тянев Д. С., Колев С. И., Йосифов В., *Метод за реализация на апаратни самоуправляващи се циклически структури*, Годишник на ТУ-Варна, Юбилеен сборник “45 години ТУ-Варна”, 2007, ISSN 1311-896X, стр. 130-135, <http://www.tyanev.com/home.php?lang=bg&mid=17&mod=0&sub=4> .
- [3]. Kolev St., Jossifov V., Master thesis: “*Design of asynchronous pipelines for floating point multiplication and division*”, FHTW – Berlin, Deutschland, 2007.
- [4]. Тянев Д. С., *Организация на компютъра*, Том 1, ISBN 978-954-20-0412-7, Технически университет-Варна, 2008.
- [5]. Patterson D. A., Hennessy J. L., *Computer Organization And Design*, Morgan Kaufmann Publishers, ISBN 1-55860-604-1, 2005.
- [6]. Sutherland I. E., *Micropipelines*, <http://research.sun.com/vlsi/Publications/KPDiscovered/micropipelines/cmultipipelines.pdf>
- [7]. John F. Wakerly, *Digital Design – principles and practices*, Fourth Edition, Prentice Hall, ISBN 0-13-186389-4, 2005.
- [8]. Булгаков С., Мещеряков В., Новоселов В., Шумилов Л., *Проектирование цифровых систем на комплектах микропрограммируемых БИС*, Москва, “Радио и связь”, 1984.
- [9]. Тянев Д. С., *Организация на компютъра – проектиране на логически структури*, ISBN 954-20-0259-9, ТУ-Варна, 2004.
- [10]. WEB-адрес на фирма XILINX – <http://www.xilinx.com/>
- [11]. Yau-Hwang Kuo, Shaw-Pyng Lo, *Partitioning and Scheduling of Asynchronous Pipelines*, Institute of Information Engineering, National Cheng Kung University, Tainan, Taiwan., Proc. of CompEuro’92, The Nitherlands, May, 1992.