

ПРИНЦИП НА ФУНКЦИОНАЛНОСТТА – ДОСТАТЪЧЕН ЛИ Е ?

Тянев Д. С.,

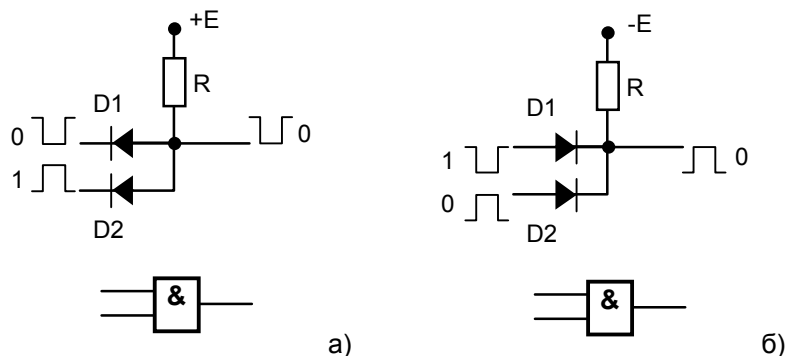
Поставя се въпроса за концептуалните разбирания, заложили като ръководно начало в стандарта за представяне логическите схеми. Според автора е спорна еднозначността и пълнотата на дефинираните графични изразни средства и правила, което е показано на различни нива и при различни случаи. Само този факт е достатъчен да се мотивира изследване, целящо усъвършенстването на този стандарт.

1. Принцип на функционалността

Представянето на инженерните проекти изисква използването на общо приети (стандартизирани) системи от правила за построяване на необходимите условни графични означения. Тук ще коментираме нормите за съставяне на условни графични означения на логически схеми.

А) Ниво електронни схеми

Нека разгледаме двете електронни схеми от фигура 1 при едни и същи работни условия. Еднаквостта на условията се състои в това, че логическите стойности ("0", "1") на входните и изходните сигнали и в двете схеми се интерпретират (кодират) в един и същи смисъл, а именно: логическа единица е нивото на захранващото напрежение (+E за рисунка а) и -E за рисунка б)), а логическата нула е нивото на общата точка.

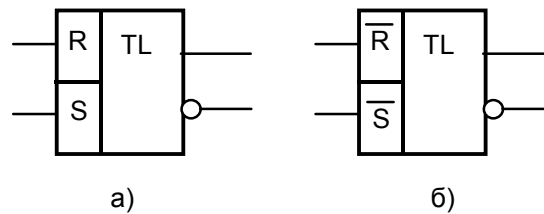


Фиг. 1 Означаване на електронната схема според логическата ѝ функция

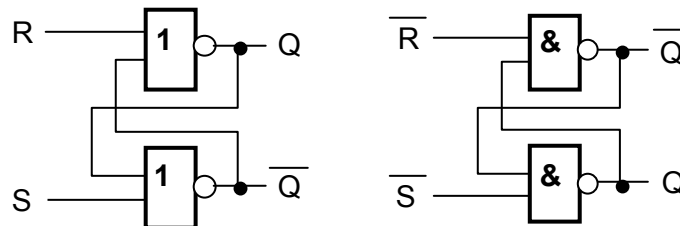
При това кодиране и според електронното функциониране на схемите, получаваният от тях логически резултат и в двата случая съответства на логическата функция "И". Ето защо, означавайки логическата функция на техническата реализация според правилата на БДС, и в двата случая условното графично означение се получава едно и също. Нека подчертаем този факт: едно и също означение – две различни технически реализации.

Б) Ниво логически елементи

За да засилим впечатлението от направеното наблюдение ще разгледаме още няколко примера. На фигура 2 са представени УГО на два тригера. От рисунка а) разбираме, че е представен RS-тригер (структура *Latch*) с право управление, т.е. превключващ се по логическите единици, а от рисунка б), че е представен RS-тригер (структура *Latch*) с инверсно управление, т.е. превключващ се по логическите нули.

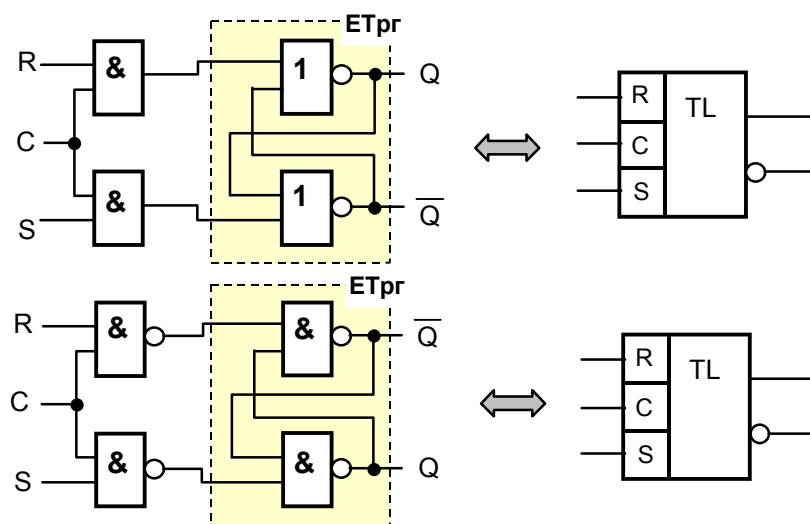
Фиг. 2 Условно графично означение на елементарни *Latch* тригери

Разликата между означенията е минимална и тя може да бъде лесно отстранена, ако се смени кодировката на логическите стойности. В същото време принципната логическа реализация на тригерите, показана на фигура 3, е съществено различна.

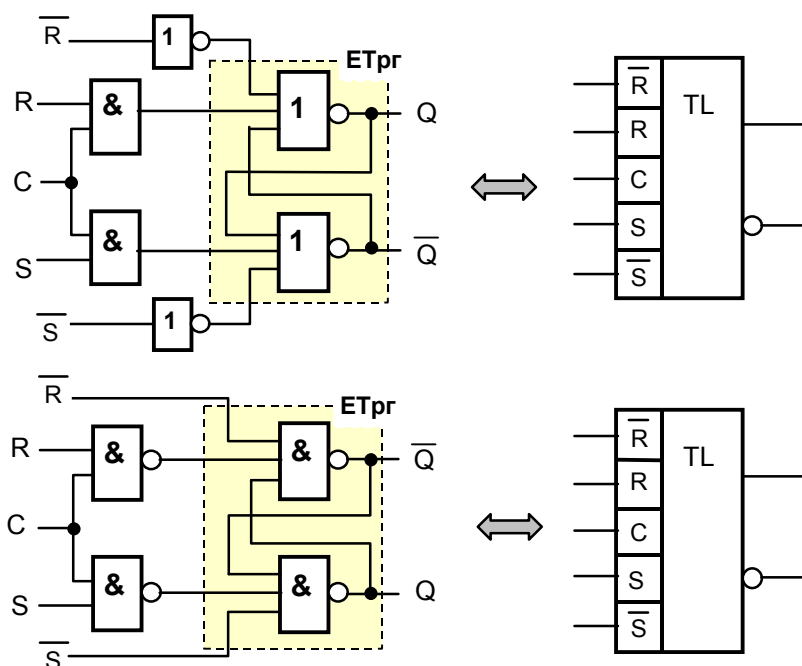
Фиг. 3 Логическа структура първични *Latch* тригери

УГО от фигура 2 са получени след обобщение единствено на логическото функциониране на схемите от фигура 3, т.е. изразяват поведението на схемите откъм техните изходи.

Споменатите по-горе тригери се превръщат в синхронни, когато им се добави специален вход за управление акта на запис, както е показано на фигура 4. Както уважаемият читател вече забелязва, УГО на тригерите от фигура 4 са напълно еднакви, но те съответстват на различни вътрешни структури, което е много показателно и следва да се има предвид при изучаването и употребяването на реалните логически елементи.

Фиг. 4 Реализация на синхронен вход в първични *Latch* тригери

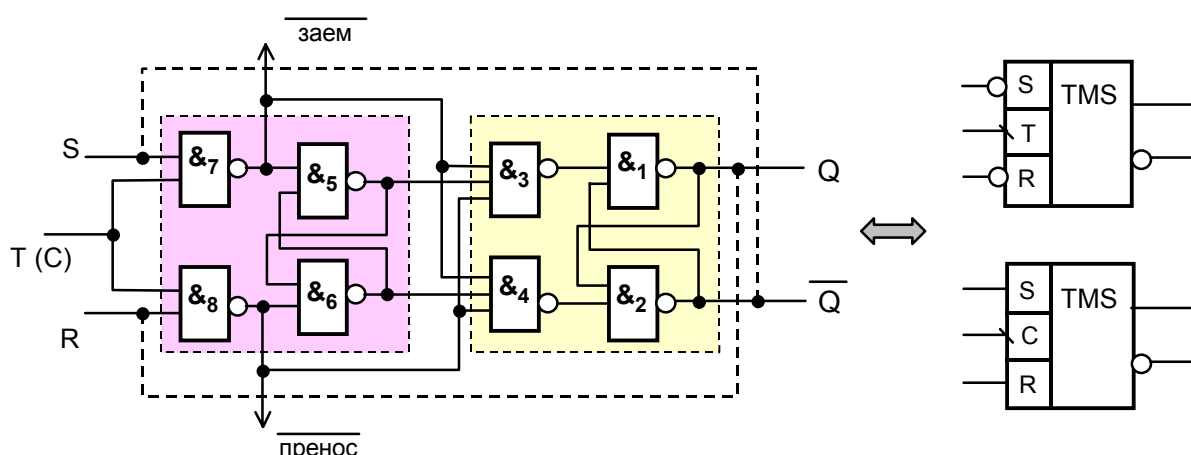
На фигура 5 са представени логически схеми на RS-тригери, допълнени с приоритетни установъчни входове. За нормална работа на тригерните схеми на тези входове трябва да бъдат осигурени във времето неактивни логически стойности. Отново се вижда, че УГО на тези елементи са еднакви, но техните вътрешни схеми – не.



Фиг. 5 Реализация на установъчни входове в първичните тригери

Следващата фигура 6 представя съвместно две тригерни структури – на Т-тригер и на RS-тригер. Приоритетните инверсни входове за установяване на тригерната схема не са изобразени. В УГО на Т-тригера обаче те са показани. Обратните връзки в логическата схема от фигура 6, които всъщност реализират Т-тригера, са показани с пунктирна линия. Освен това са показани бързодействащите изходи на тригера за “пренос” и за “заем”, които се използват при включването му в многоразрядни броячни схеми. Поради това, че функциите за пренос и за заем се генерират в инверсна стойност, включването им към Т-входовете на съседните разряди в броячните схеми, ще изисква допълнителен инвертор.

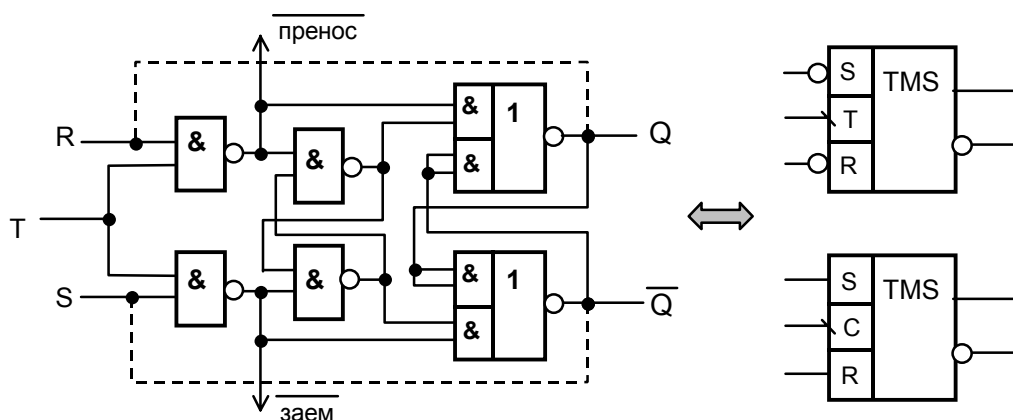
Когато обратните връзки в разглежданата схема отсъстват, а входните логически схеми И-НЕ са допълнени, както е показано, с S и R входове, тригерната схема представлява синхронен RS-тригер, тип MS със забраняващи връзки и тогава входът Т следва да се обозначи с буква С, както е показано в долното УГО на фигура 6.



Фиг. 6 Асинхронен MS Т-тригер и синхронен MS RS-тригер със забраняващи връзки

Времето за превключване на тригерните схеми със забраняващи връзки може да бъде намалено чрез намаляване на броя на логическите инвертори. За целта при синтеза на логическата схема трябва да се използват различни логически бази за логическите елементи на елементарните тригери. Този подход е показан на фигура 7, където основният елементарен тригер е в базиса И-НЕ, а допълнителният – в базиса И-ИЛИ-НЕ. Тригерната

схема отново показва съвместно изобразяване на два тригера – асинхронен MS T-тригер със забраняващи връзки и синхронен MS RS-тригер със забраняващи връзки. Обратните връзки в схемата са изобразени с пунктирни линии. И в тази схема, в случай, че бъде разглеждан T-тригер, са показани бързодействащите изходи на функциите за “пренос” и за “заем”, които се използват при синтез на многоразрядни броячни схеми.

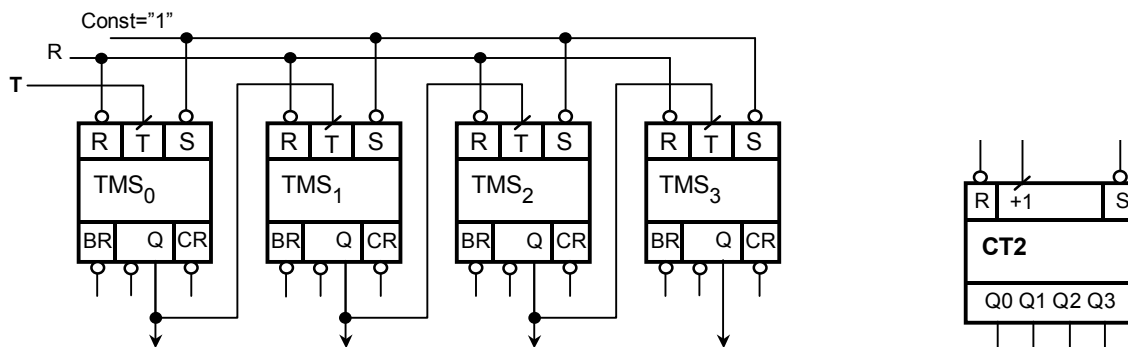


Фиг. 7 T-тригер и RS-тригер със забраняващи връзки и различни логически елементи

В логическата схема на фигура 7 не са изобразени приоритетните установъчни входове на тригера. Обясненията, касаещи изобразените на тази фигура УГО, съответстват по аналогичен начин на изложените на фигура 6.

В) Ниво логически възли

Нека сега се издигнем от нивото на логическите елементи към нивото на логическите възли. Ще разгледаме логическата схема на сумиращ 4-битов двоичен брояч с последователен пренос, (виж фигура 8), построен с помощта на разгледания вече асинхронен T-тригер.

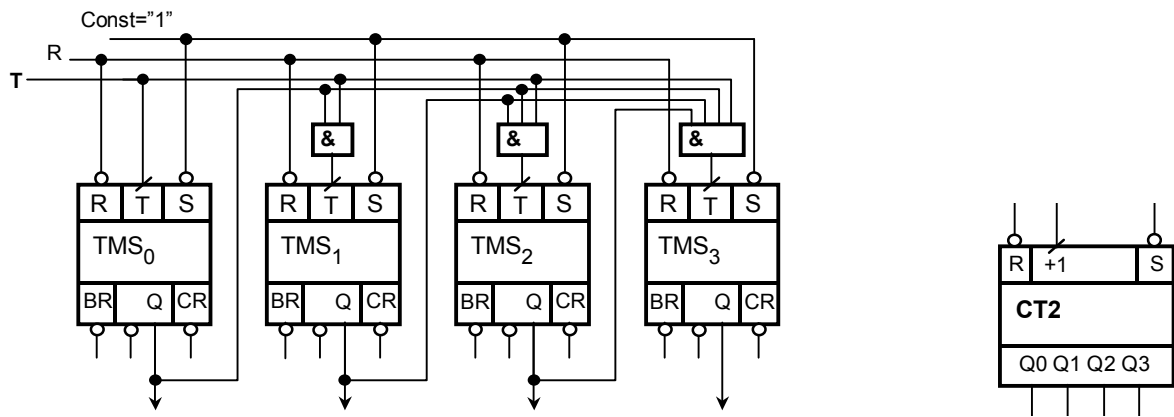


Фиг. 8 Принципна логическа схема на 4-битов брояч с последователен пренос и неговото УГО

От схемата се вижда, че броячът се реализира, като правите изходи на тригерите се използват в качеството си на входни сигнали към по-старшите разряди. Останалите изходи на тригерите не се използват. За да се осигури работното състояние, всички установъчни входове S са обединени и на тях се подава логическа константа “1”. Всички R входове са също обединени за да се постига едновременно нулиране съдържанието на тригерите, т.е. на брояча. Като вход на брояча се явява T-входът на най-младшия тригер. Всеки единичен импулс, постъпил в брояча води до неговото превключване в ново състояние. Състоянията са 16 на брой ($2^4=16$) – от комбинация 0000 до комбинация 1111.

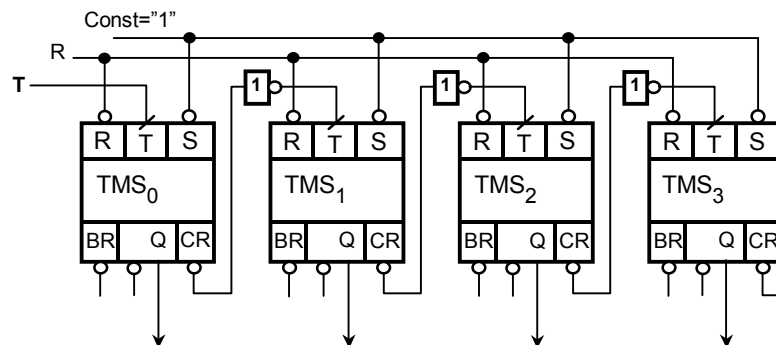
За да поясним обобщаващите възможности на стандарта, принципната логическа схема е допълнена с представящото я УГО. И тук УГО е получено от гледната точка на изходната функция.

На фигура 9 е представена принципната логическата схема на двоичен брояч с паралелен пренос и неговото УГО.



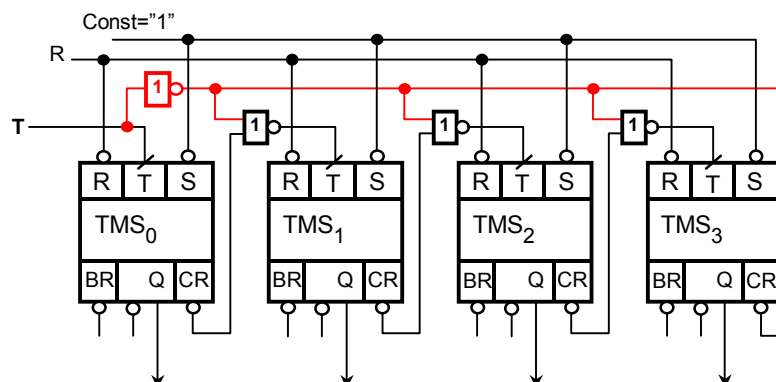
Фиг. 9 Принципна логическа схема на 4-битов брояч с паралелен пренос

По-долу на фигура 10 е представена принципната логическа схема на сумиращ двоичен брояч с последователен пренос, в която са използвани изходите CR на тригерите. Тези изходи са допълнени с инвертори за да се получи правата стойност на функциите на преноса във всеки отделен разряд.



Фиг. 10 Принципна логическа схема на 4-битов брояч с последователен пренос с използване на изходите за ускорен пренос CR

С цел да се намали времето за превключване на брояча, последователно разпространяващите се преноси могат да бъдат стробирани от входния броячен импулс Т, както е показано на фигура 11. Стробирането се постига чрез логическата функция:

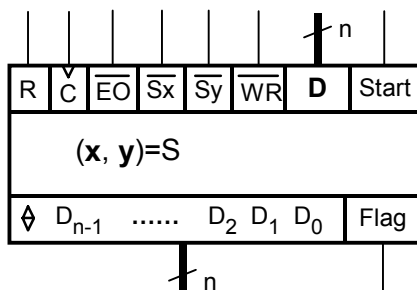


Фиг. 11 Стробиране на последователните преноси

Учудващото тук е това, че и за тези две схеми се получава същото УГО както показаното за предишните две.

В) Ниво логически устройства

Нивото на логическите устройства, определени според [1], ще илюстрираме чрез пример, представен в [2], където е проектирано устройство за скалярно умножение на два вектора. Устройството е проектирано в два варианта – според синхронния и асинхронния методи за управление. Въпреки съществените различия в принципните логически схеми на двата варианта, условното им графично означение, получено там според правилата на стандарта, е едно и също (виж фигура 12).



Фиг. 12 Функционално означение на устройство за скалярно умножение на два вектора

Принципът на функционалността, който беше илюстриран по-горе, води до същите ефекти и на ниво системи. Колкото по-висока е степента на обобщение в УГО, толкова по-невидими стават различията, които се съдържат в реалните принципни схеми.

Извод:

Ако трябва да бъдат изразени само логическите функции на логическите схеми принципът на функционалността би бил достатъчен, но като имаме предвид, че логическите схеми се проектират за да бъдат реализирани с реални градивни елементи, считаме, че този принцип е само необходим, но не и достатъчен за тяхното разчитане.

2. Какво да се прави?

Стандартът е факт и той се използва. Заложеният в неговите правила принцип на функционалността има своите последствия – добри или лоши. Въпросът, който искаме да повдигнем тук, е възможно ли е да обогатим тези правила така, щото условните графични означения да представят логическите схеми по-точно от към тяхната конкретна реализация?

Литература

- [1]. Тянев Д., *Електронни цифрови машини*, ISBN – 954-20-0016-2, Технически Университет – Варна, 1995 г.
- [2]. Тянев Д., *Организация на компютъра – ръководство за проектиране на логически структури*, ISBN – 954-20-0120-7, Технически Университет – Варна, 2000 г.