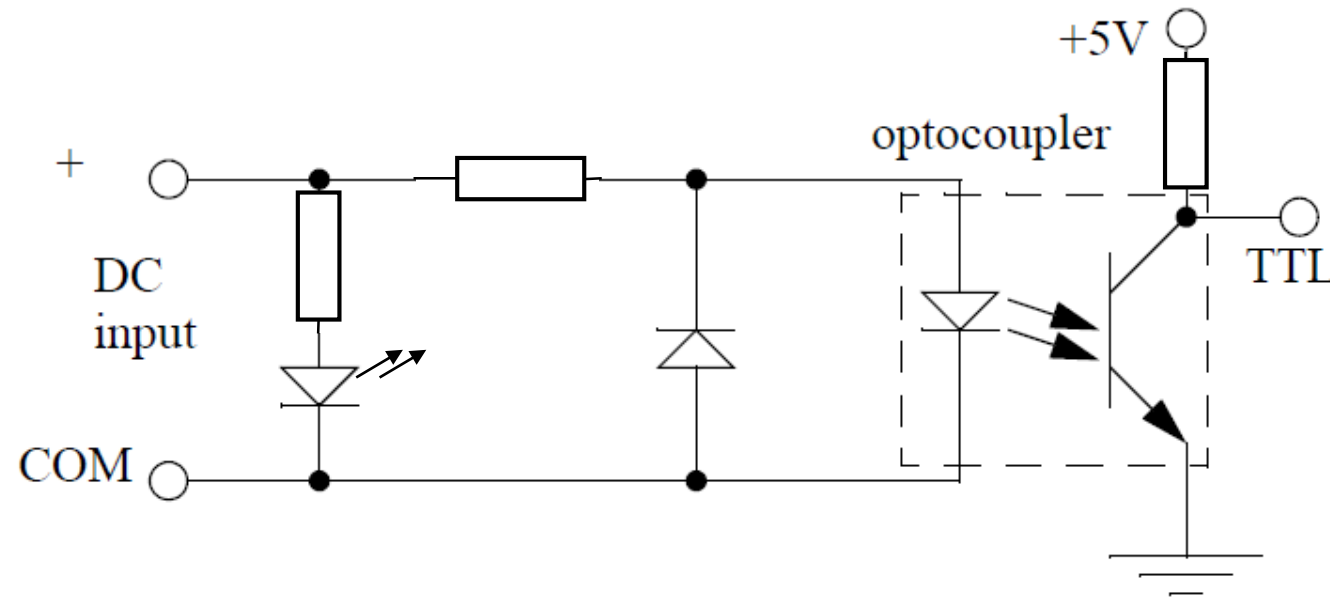


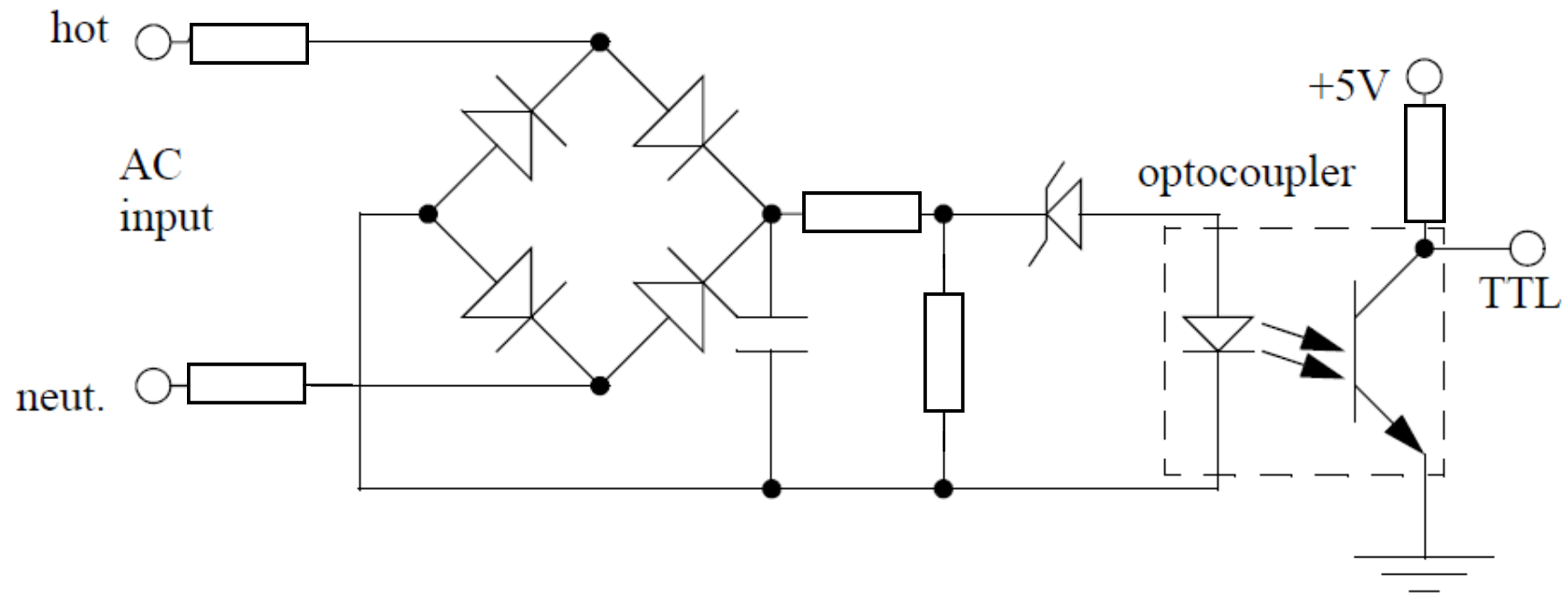
Схемни решения на цифрови (релейни) входни блокове. Входните вериги на PLC могат да се класифицират по няколко основни признака, най-общият е според вида на напрежителния сигнал на който реагират – постояннотокови и променливотокови входове. Захранващият блок на контролера обикновено не захранва входовете. Затова трябва да се осигури външен източник, който да подава напрежение към тях. Входните вериги на PLC трябва да транслират (преобразуват) различни логически нива до 5Vdc логически нива, най-често използвани в контролерите.

По принцип схемите формират входен сигнал за задействане на оптрон. По този начин се осигурява електрическа изолация на външните електрически вериги от вътрешните вериги. Други схемни компоненти се използват, за да се предпази схемата от пренапрежение или от обръщане на посоката на напрежението. Това може да се направи със схемни решения, които са показани по-долу.

Схемни решения на цифрови (релейни) входни блокове. Входна схема, реагираща при подаване на постоянно напрежение.



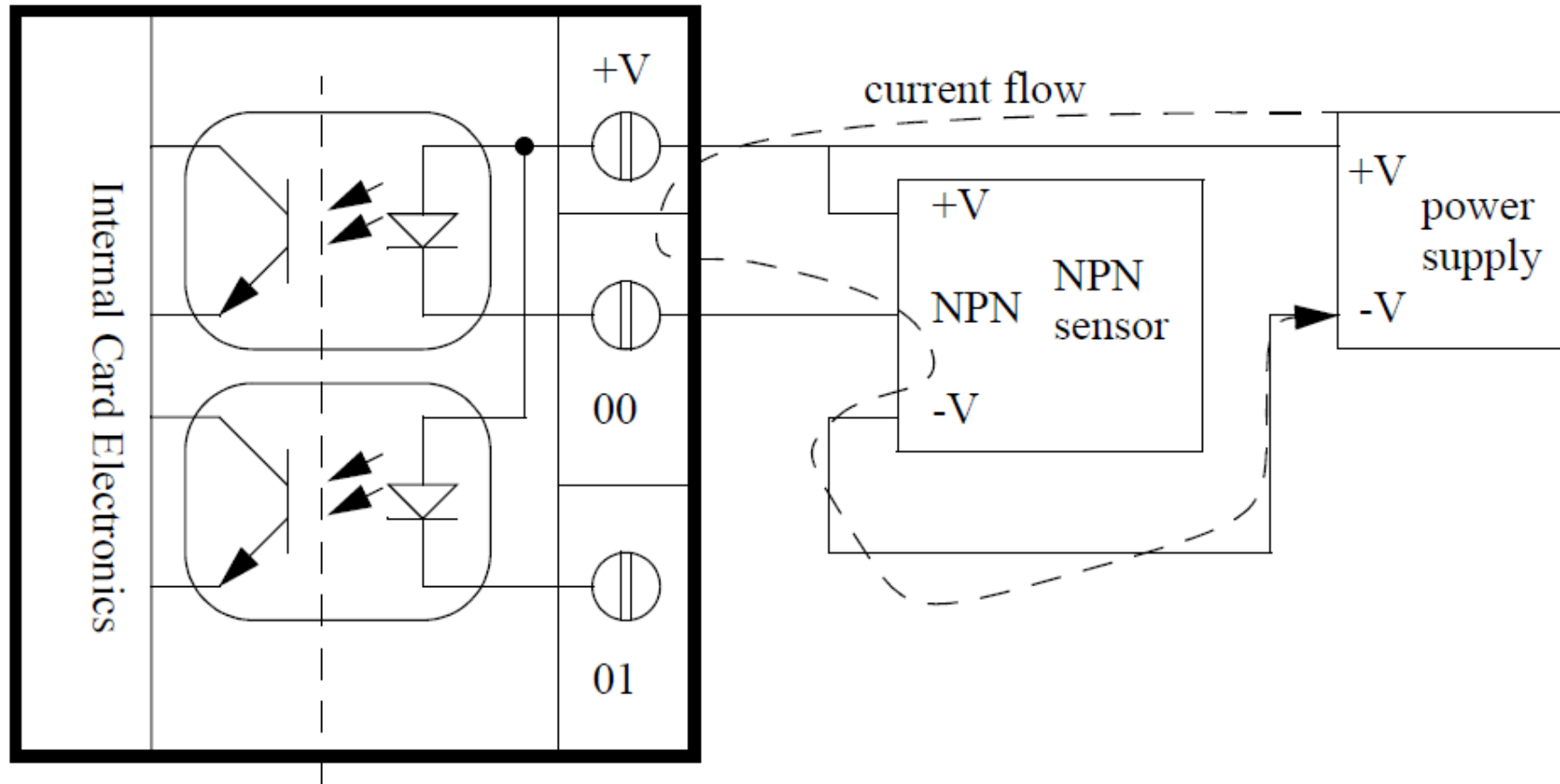
Схемни решения на цифрови (релейни) входни блокове. Входна схема, реагираща при подаване на променливо напрежение.



Схемни решения на цифрови (релейни) входни блокове. Друг признак е според начина на свързване на общия входящ проводник (към нулевия или към захранващия проводник). Когато се разглежда действието на сензора като източник или като консуматор, това се отнася за изхода на сензора, който действа като ключ, свързан към цифровия вход на контролера. Изходът на сензора обикновено е транзистор, който ще действа като ключ (с известна загуба на напрежение). При разглеждане на сензорите терминът източник често се заменя с PNP, а консуматор - с NPN.

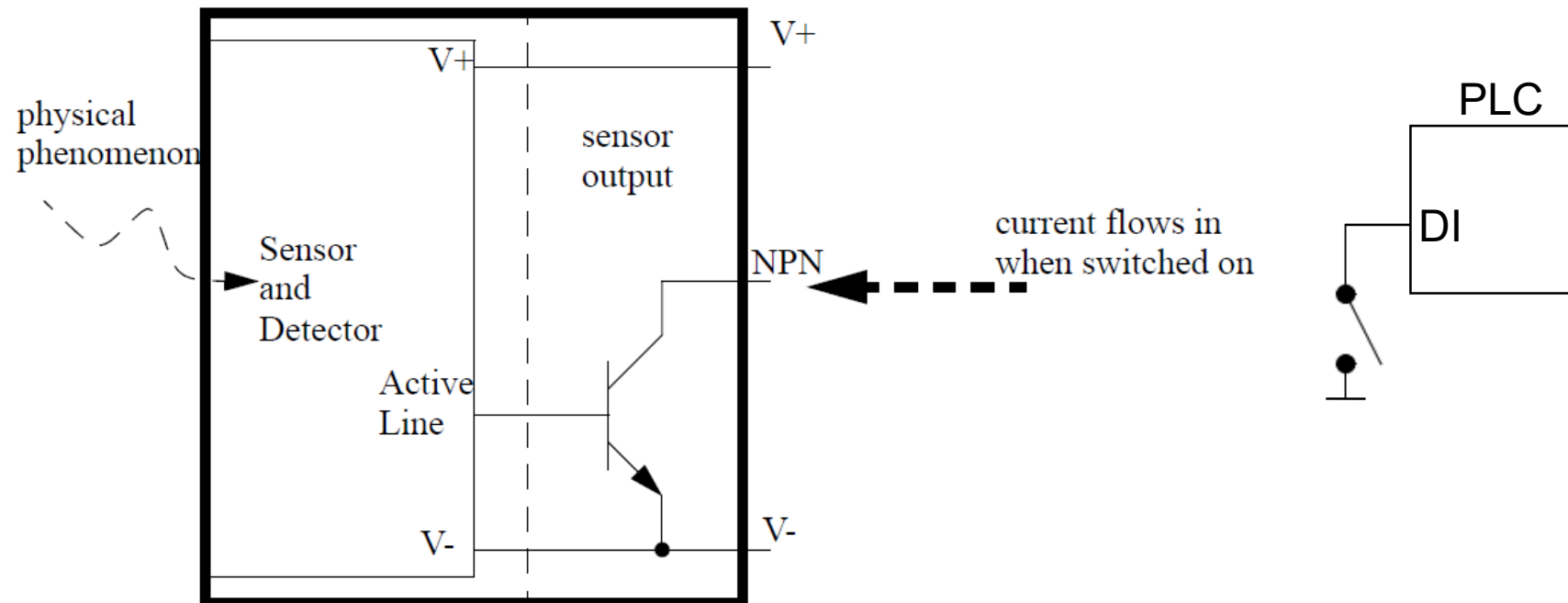
PNP транзистор се използва за изход, работещ като източник. При този тип изход токът тече от захранващия източник през изхода навън от сензора. Тогава общ входящ проводник е нулевият проводник. NPN транзистор се използва за изход, работещ като консуматор. В този случай токът тече към сензора, през изхода към нулевия проводник. Тогава общ входящ проводник е захранващият проводник.

Схемни решения на цифрови (релейни) входни блокове

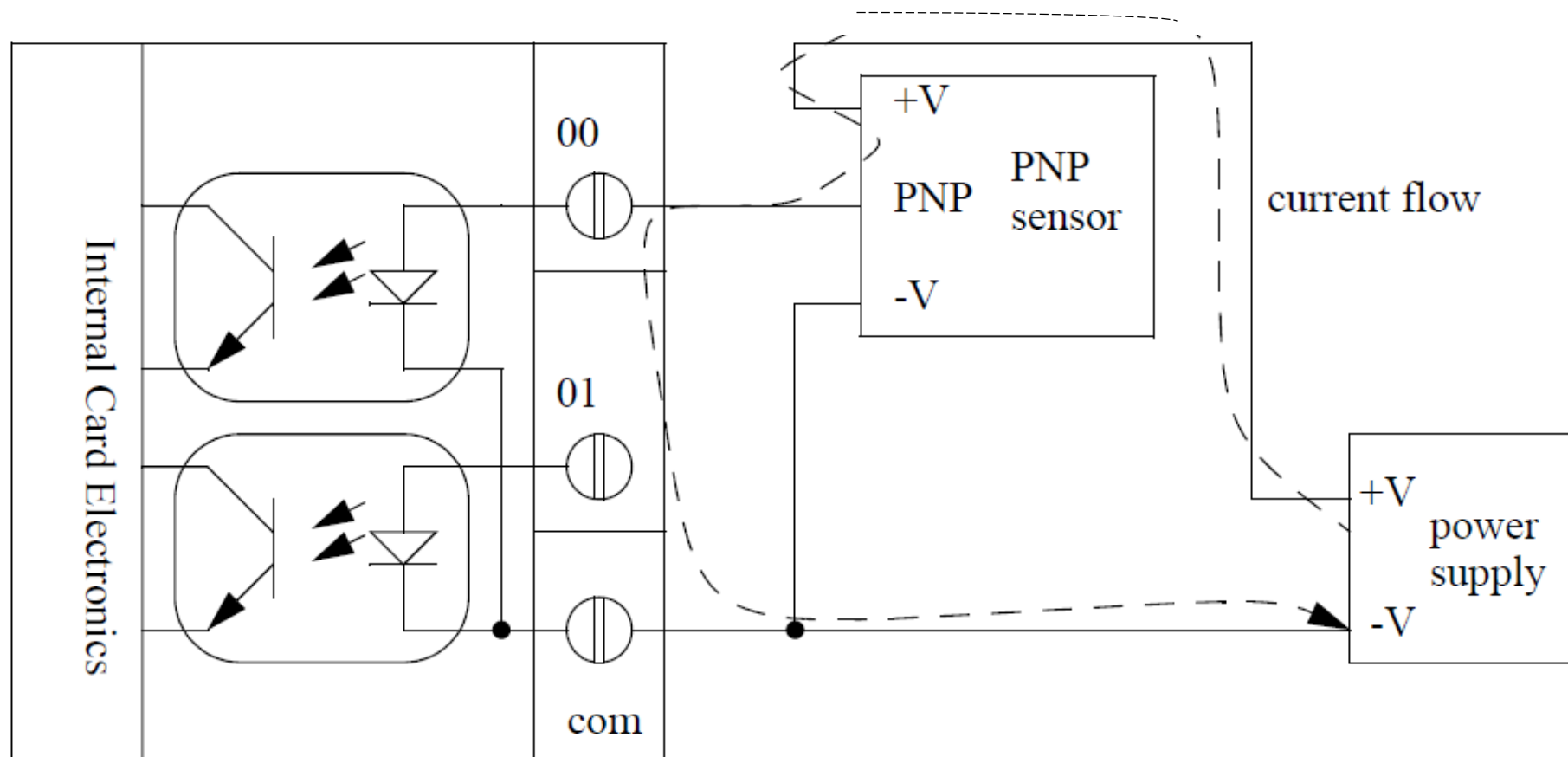


Входни вериги на PLC за свързване към сензори от NPN тип

Схемни решения на цифрови (релейни) входни блокове. Показан е опростен пример на сензор с изход, работещ като консуматор. Сензорът се нуждае от захранващо напрежение, за да работи и трябва да бъде свързан към захранващ източник. Ако сензорът е открил промяна в състоянието на обекта, ще се задейства активния изход. Той е свързан директно с базата на NPN транзистор. Тогава той ще се отпусти токът ще протече през изхода на сензора към нулевия проводник.

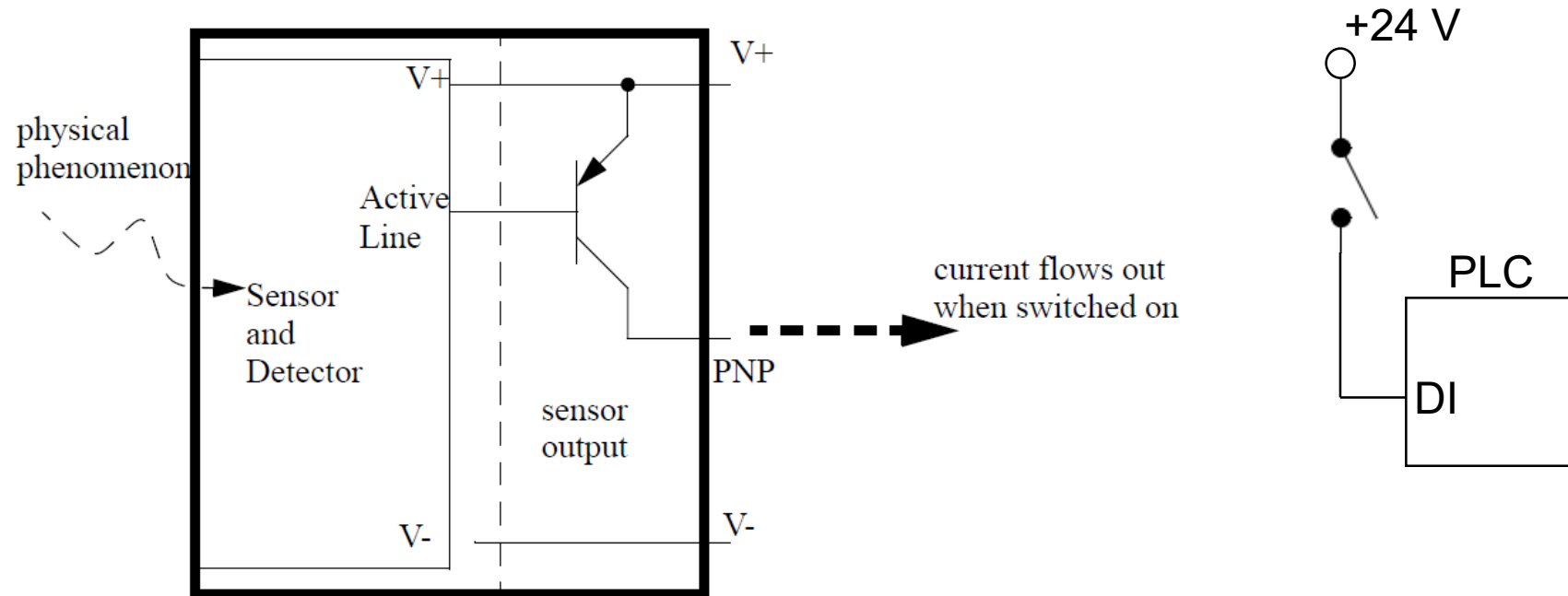


Схемни решения на цифрови (релейни) входни блокове



Входни вериги на PLC за свързване към сензори от PNP тип

Схемни решения на цифрови (релейни) входни блокове. Показан е опростен пример на сензор с изход, работещ като източник. Сензорът се нуждае от захранващо напрежение, за да работи и трябва да бъде свързан към захранващ източник. Ако сензорът е открил промяна в състоянието на обекта, ще се задейства активния изход. Той е свързан директно с базата на PNP транзистор. Тогава той ще се отпусне и токът ще протече от захранващия източник през изхода навън от сензора.



Схемни решения на цифрови (релейни) входни блокове. В CPU модула на S7-1200 са предвидени цифрови входни филтри, които предпазват програмата от грешно сработване в следствие на нежелани бързи промени на входните сигнали, което може да е резултат от разтрептяване на контакт или електрически шум. По подразбиране времезакъснението на филтъра е 6,4 ms и се блокират нежеланите преходи от типичните механични контакти. В различни части на програмата може да се изисква по-кратко време за филтриране, за да се откриват промени и да се реагира на входни сигнали от бързи сензори, или по-дълги времена на филтриране, за да се блокират бавни контакти или по-нискочестотен шум.

Времезакъснението на входния филтър от 6,4 ms означава, че за да се регистрира промяна на един сигнал от "0" на "1" или от "1" на "0", трябва новото ниво да се задържи за около 6,4 ms, и че единичен положителен или отрицателен импулс, по-къс от около 6,4 ms няма да бъде регистриран.

Всяка входна точка има самостоятелна конфигурация на филтъра, която се прилага за всички приложения - четене на входове, прекъсвания, „pulse catch“ и високоскоростни броячни входове.

Обхвати на аналогови входни блокове на програмируемия логически контролер S7-1200

System		Voltage Measuring Range						
Decimal	Hexadecimal	±10 V	±5 V	±2.5 V	±1.25V		0 to 10 V	
32767	7FFF	11.851 V	5.926 V	2.963 V	1.481 V	Overflow	11.851 V	Overflow
32512	7F00							
32511	7EFF	11.759 V	5.879 V	2.940 V	1.470 V	Overshoot range	11.759 V	Overshoot range
27649	6C01							
27648	6C00	10 V	5 V	2.5 V	1.250 V	Rated range	10 V	Rated range
20736	5100	7.5 V	3.75 V	1.875 V	0.938 V		7.5 V	
1	1	361.7 µV	180.8 µV	90.4 µV	45.2 µV		361.7 µV	
0	0	0 V	0 V	0 V	0 V		0 V	
-1	FFFF						Negative values are not supported	
-20736	AF00	-7.5 V	-3.75 V	-1.875 V	-0.938 V			
-27648	9400	-10 V	-5 V	-2.5 V	-1.250 V			
-27649	93FF							
-32512	8100	-11.759 V	-5.879 V	-2.940 V	-1.470 V	Undershoot range		
-32513	80FF					Underflow		
-32768	8000	-11.851 V	-5.926 V	-2.963 V	-1.481 V			

Обхвати на аналогови входни блокове на програмируемия логически контролер S7-1200

System		Current measuring range		
Decimal	Hexidecimal	0 mA to 20 mA	4 mA to 20 mA	
32767	7FFF	23.70 mA	22.96 mA	Overflow
32512	7F00			
32511	7EFF	23.52 mA	22.81 mA	Overshoot range
27649	6C01			
27648	6C00	20 mA	20 mA	Nominal range
20736	5100	15 mA	16 mA	
1	1	723.4 nA	4 mA + 578.7 nA	
0	0	0 mA	4 mA	
-1	FFFF			
-4864	ED00	-3.52 mA	1.185 mA	Undershoot range
-4865	ECFF			
-32768	8000			Underflow

Схемни решения на цифрови (релейни) изходни блокове. PLC изходите трябва да транслират (преобразуват) логическите нива на шината за данни на PLC с напрежение 5Vdc към стойностите на външния източник на напрежение. Това може да се направи със схеми, подобни на тези, показани по-долу.

По принцип схемите използват оптрон за превключване на външните вериги. Това осигурява електрическа изолация на външните електрически вериги от вътрешните вериги. Други схемни компоненти се използват, за защита от пренапрежение или от обръщане на посоката на напрежението.

Сухи контакти - отделно реле е предназначено за всеки изход. Това позволява превключване на различни напрежения (променливи или постоянни със стойности до максимално допустимите), както и галванично разделяне от другите изходи и от PLC. Този метод е най-малко чувствителен към измененията на напрежението и отскоците.

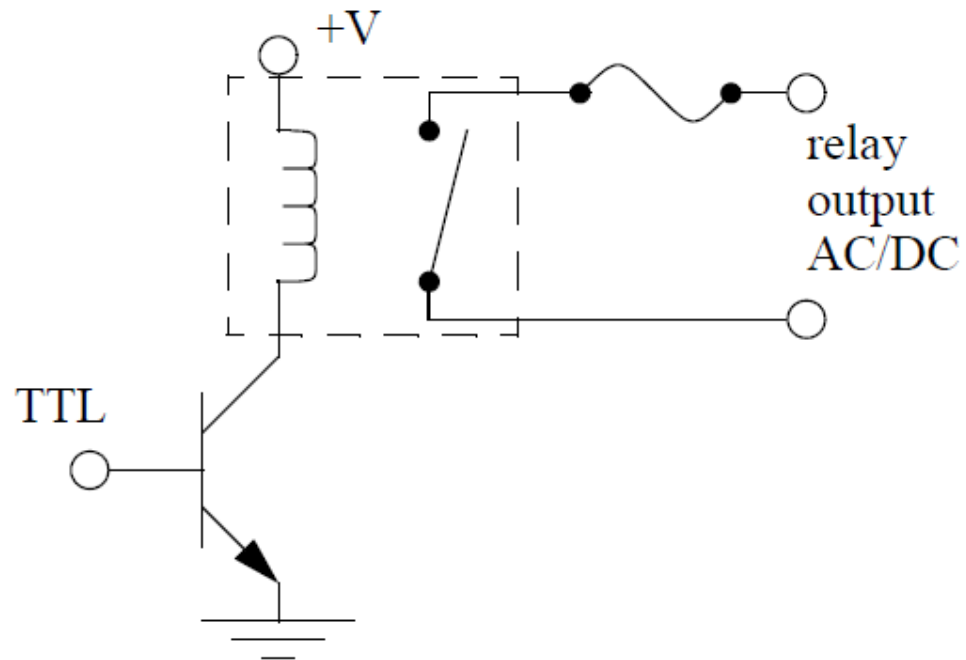
Комутирани изходи - към PLC картата се подава напрежение и то се превключва към различни изходи с използване на безконтактни елементи - транзистори, триаци и т.н. Времето им на реакция е много по-малко от 1 ms.

Схемни решения на цифрови (релейни) изходни блокове. При управление на индуктивни товари трябва да се използват потискащи (супресорни вериги), за да се ограничи повишаването на напрежението, когато управляващ изход на контролера се изключва. Потискащите вериги предпазват изходите от повреда, причинена от високите стойности на преходното напрежение, което възниква, когато токът през индуктивен товар е прекъснат.

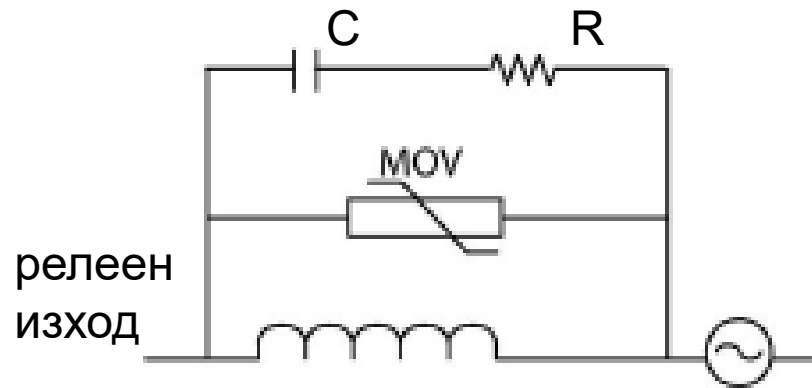
В допълнение, супресорните вериги ограничават електромагнитните смущения, генерирани при превключване на индуктивен товар. Високочестотният шум от лошо потиснати индуктивни товари може да наруши работата на PLC. Поставяне на външна супресорна верига, така че тя да е електрически паралелна на товара и физически разположена в близост до него е най-ефективният начин за намаляване на електромагнитните смущения.

Добро решение за потискане на смущения е използването на контактори и други индуктивни товари, за които техният производител е осигурил интегрирани в товара схеми за потискане.

Схемни решения на цифрови (релейни) изходни блокове. Изходни стъпала за превключване на променливо напрежение могат да се реализират с релета или с триаци. Изходните релета чрез своите контакти може да се използват и за включване на мощни ключови елементи (контактори) за управление на изпълнителни механизми с голяма мощност. Също така контактите на релетата може да участват в реализирането на сложни логически схеми като управляват цифрови входове.



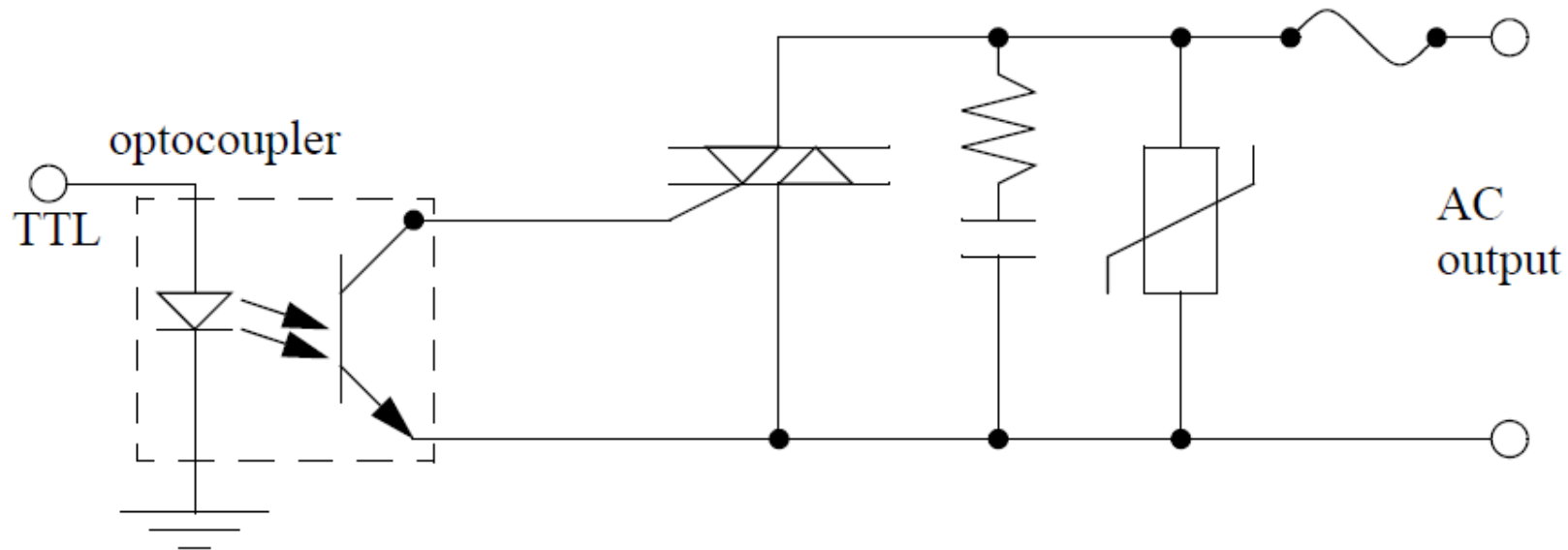
Схемни решения на цифрови (релейни) изходни блокове. При управление на индуктивни товари, които се захранват с променливо напрежение, е необходимо да се осигурят защитни елементи.



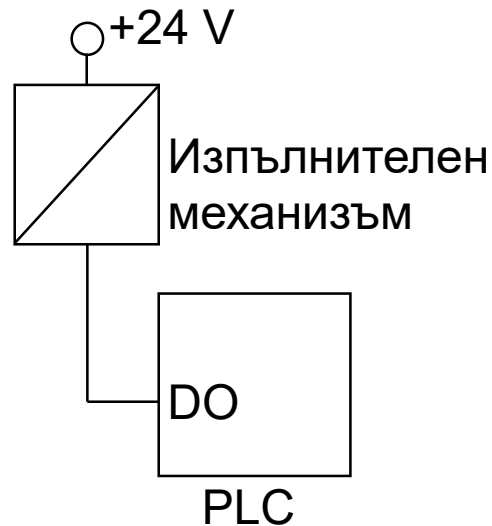
Стойностите на резистора и кондензатора се избират от показаната таблица в зависимост от стойността на товара. Работното напрежение на варистора (MOV) трябва да бъде поне с 20% по-високо от номиналното напрежение на мрежата.

I rms Amps	Inductive load		Suppressor values		
	230 V AC	120 V AC	Resistor		Capacitor
	VA	VA	Ω	W (power rating)	nF
0.02	4.6	2.4	15000	0.1	15
0.05	11.5	6	5600	0.25	470
0.1	23	12	2700	0.5	100
0.2	46	24	1500	1	150
0.5	115	60	560	2.5	470
1	230	120	270	5	1000
2	460	240	150	10	1500

Схемни решения на цифрови (релейни) изходни блокове.
 Използването на триаци изисква допълнителни схемни решения, продиктувани от принципа им на работа. Първото е синхронизиране на управлението с преминаването на променливото напрежение през нула. Също така трябва да се вземат мерки за потискането на отскоци и смущения.

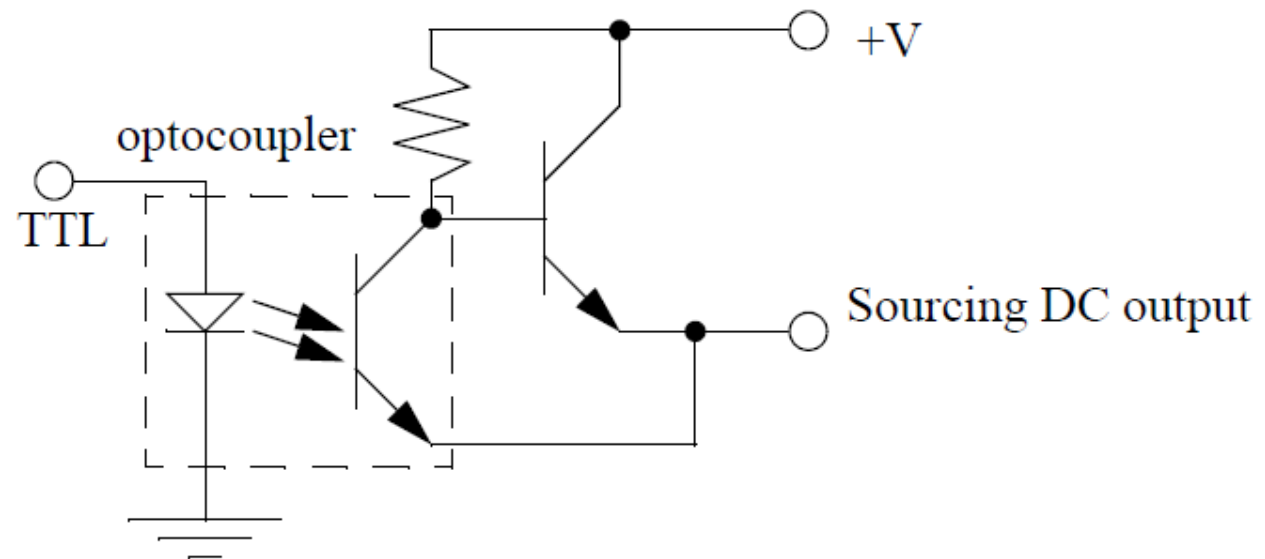
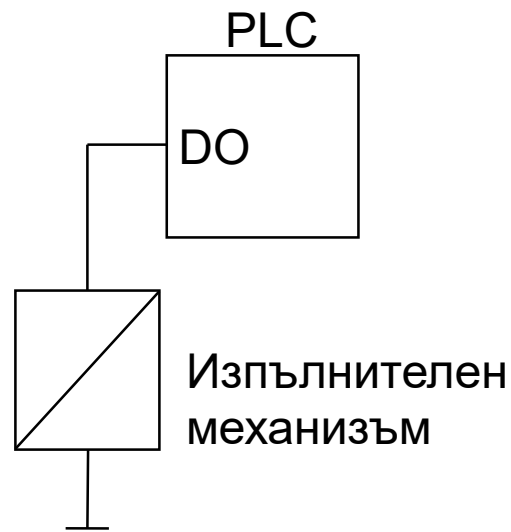


Схемни решения на цифрови (релейни) изходни блокове. Когато изходите превключват само постоянно напрежение се използват биполярни или MOS транзистори като ключови елементи. Тези изходи имат по-високо бързодействие от предните. При изходните точки също има класификация според начина на действие – източник или консуматор. Тези режими също се разглеждат съответно като PNP тип или NPN тип. От тях се определя и общия проводник – захранващият или нулевият.

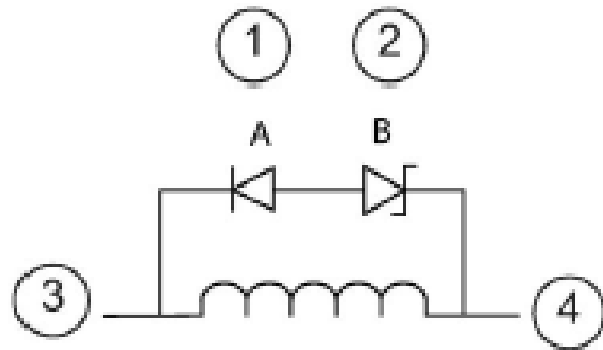


Изход, който работи в режим на консуматор (NPN тип) използва нулевия проводник като общ. При включено състояние токът протича от захранващия източник, през товара (изпълнителния механизъм) и през изхода на контролера към нула.

Схемни решения на цифрови (релейни) изходни блокове. Изход, който работи в режим на източник (PNP тип) използва захранващия проводник като общ. При включено състояние токът протича от изхода на контролера, през товара (изпълнителния механизъм) към нула.



Схемни решения на цифрови (релейни) изходни блокове. При управление на индуктивни товари, захранвани с постоянно напрежение, защитните вериги се различават от тези при променливотоковите товари.



- 1 – диод 1N4001 или еквивалентен
- 2 – ценеров диод 8,2V (DC изход)
ценеров диод 36V (релеен изход)
- 3 – захранване 24V
- 4 – изход на PLC

В повечето приложения добавянето на диод (А) в паралел на постояннотоковия индуктивен товар е подходящо, но ако приложението изисква по-кратки времена за изключване, тогава се препоръчва добавянето на ценерови диоди (В). Ценеровият диод трябва да бъде правилно оразмерен, за да отговаря на стойността на тока в изходната верига.

Обхвати на аналогови изходни блокове на програмируемия логически контролер S7-1200

System		Voltage Output Range		
Decimal	Hexadecimal	± 10 V		
32767	7FFF	See note 1	Overflow	
32512	7F00	See note 1		
32511	7EFF	11.76 V	Overshoot range	
27649	6C01			
27648	6C00	10 V	Rated range	
20736	5100	7.5 V		
1	1	361.7 μ V		
0	0	0 V		
-1	FFFF	-361.7 μ V		
-20736	AF00	-7.5 V		
-27648	9400	-10 V		
-27649	93FF			Undershoot range
-32512	8100	-11.76 V		
-32513	80FF	See note 1		Underflow
-32768	8000	See note 1		

Обхвати на аналогови изходни блокове на програмируемия логически контролер S7-1200

System		Current Output Range	
Decimal	Hexadecimal	0 mA to 20 mA	
32767	7FFF	See note 1	Overflow
32512	7F00	See note 1	
32511	7EFF	23.52 mA	Overshoot range
27649	6C01		
27648	6C00	20 mA	Rated range
20736	5100	15 mA	
1	1	723.4 nA	
0	0	0 mA	

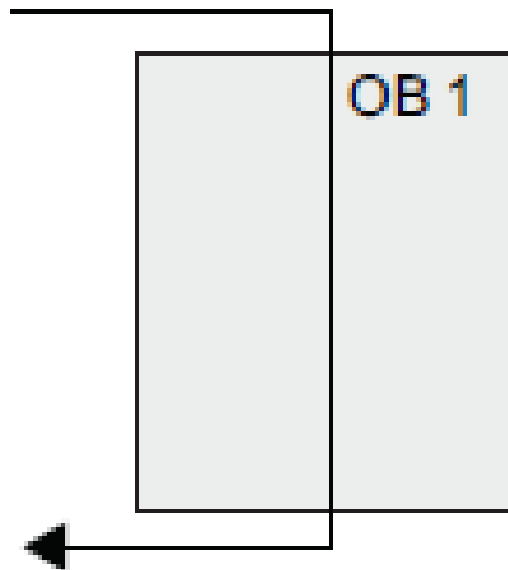
Структура на потребителската програма. Според изискванията на приложението, може да се избере линейна структура или модулна структура за създаване на потребителската програма:

- Линейната програма изпълнява всички инструкции на задачите за автоматизация последователно, една след друга. Обикновено линейната програма поставя всички инструкции на кода в един ОВ за циклично изпълнение на програмата (ОВ 1).

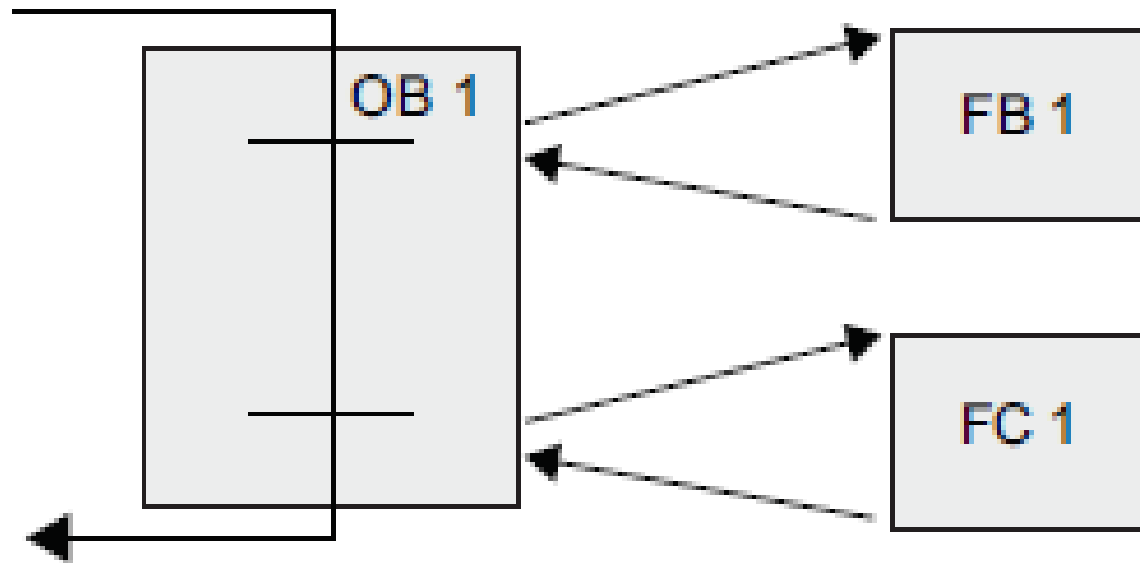
- Модулната програма извиква специфични блокове код, които изпълняват специфични задачи. За да се създаде модулна структура, сложната задача за автоматизация се разделя на по-малки подчинени задачи, които съответстват на технологичните функции на процеса. Всеки блок код осигурява програмния сегмент за всяка подчинена задача. Програмата се структурира чрез извикване на даден блок код от друг блок.

Чрез създаване на типови блокове код, които могат да бъдат използвани повторно в рамките на потребителската програма, може да се опрости проектирането и изпълнението на потребителската програма.

Структура на потребителската програма



Линейна
структура



Модулна
структура

Структура на потребителската програма

Използването на генерични (типови) блокове код има следните предимства:

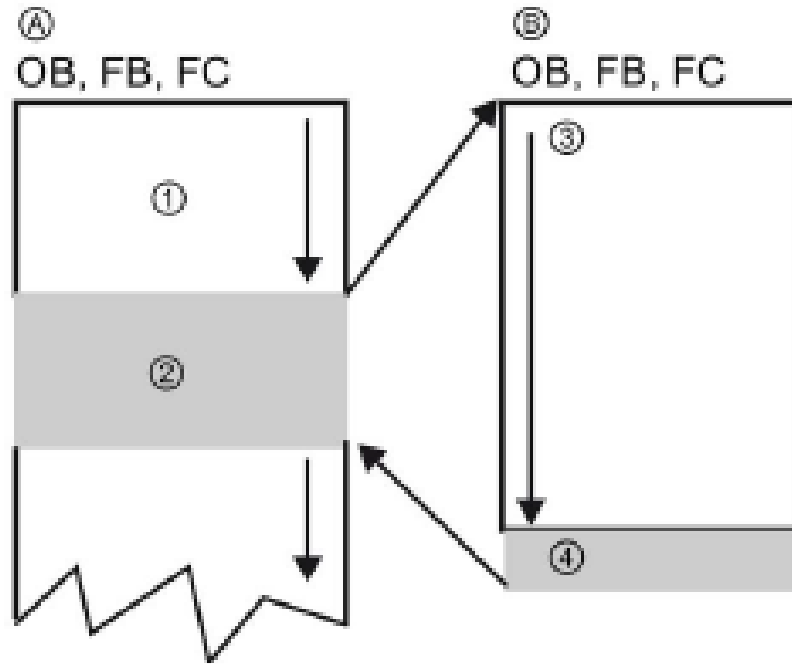
- Може да се създават блокове с код за многократна употреба за стандартни задачи, като например за управление на помпа или двигател. Може също така тези типови блокове код да се съхраняват в библиотека, от която да се използват за различни приложения или решения.

- Когато потребителската програма се структурира чрез използване на модулни компоненти, които са свързани с функционалността на задачите, проектирането на потребителската програма става по-лесно за разбиране и управление. Модулните компоненти не само помагат за стандартизиране на проектирането на програмата, но също така помагат актуализирането или модифицирането на програмния код да се извърши по-бързо и по-лесно.

- Създаването на модулни компоненти опростява отстраняването на грешки в програмата. Структурирането на пълната програма като набор от модулни сегменти позволява да се тества функционалността на всеки блок код, когато е разработен.

Структура на потребителската програма

Когато един блок код извиква друг блок код, процесорът изпълнява програмния код в извикания блок. След като изпълнението на извикания блок код е завършено, процесорът възобновява изпълнението на инструкциите от повикващия блок. Обработката продължава с изпълнението на инструкцията, която следва извикването.



А – извикващ блок

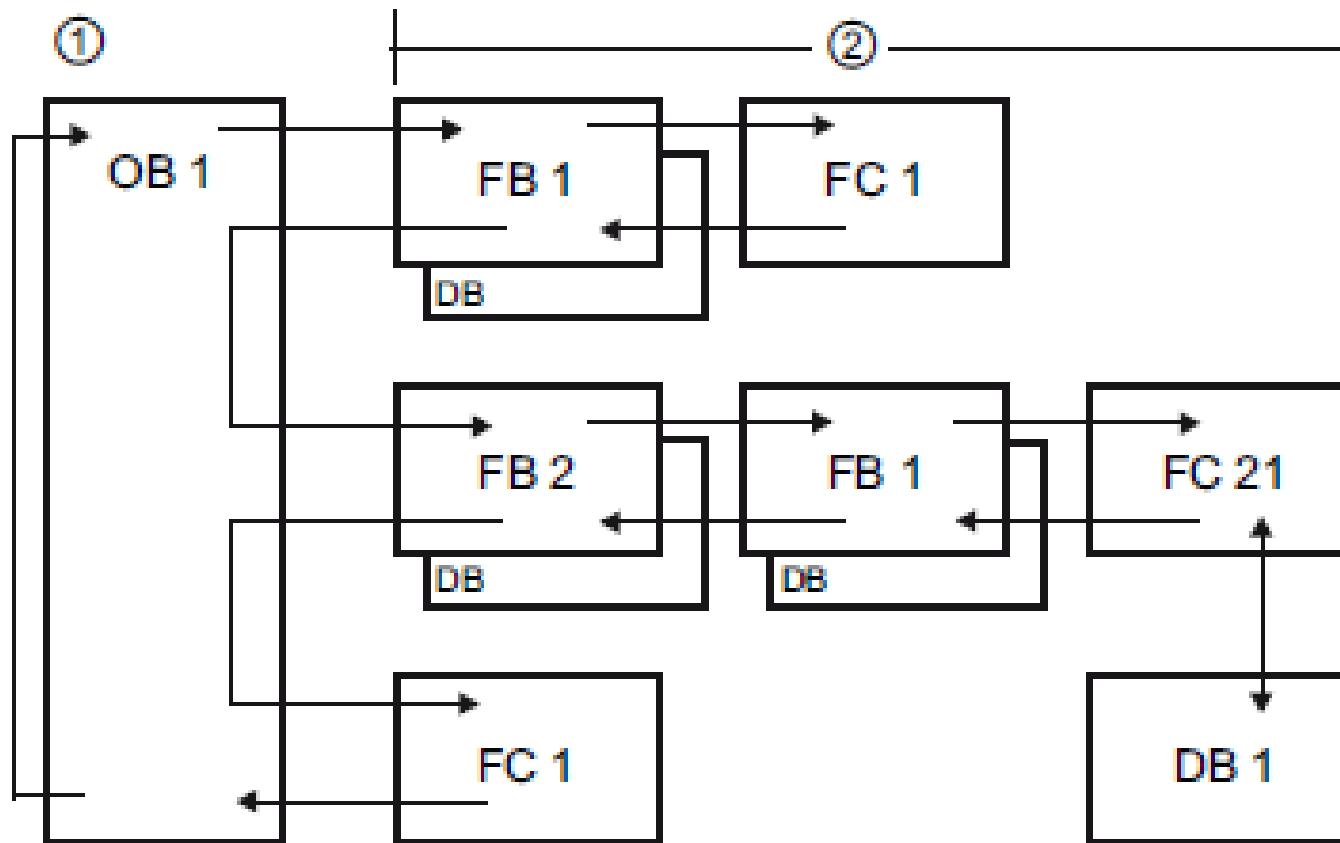
В – извикан (или прекъсващ) блок

① – изпълнение на програмата

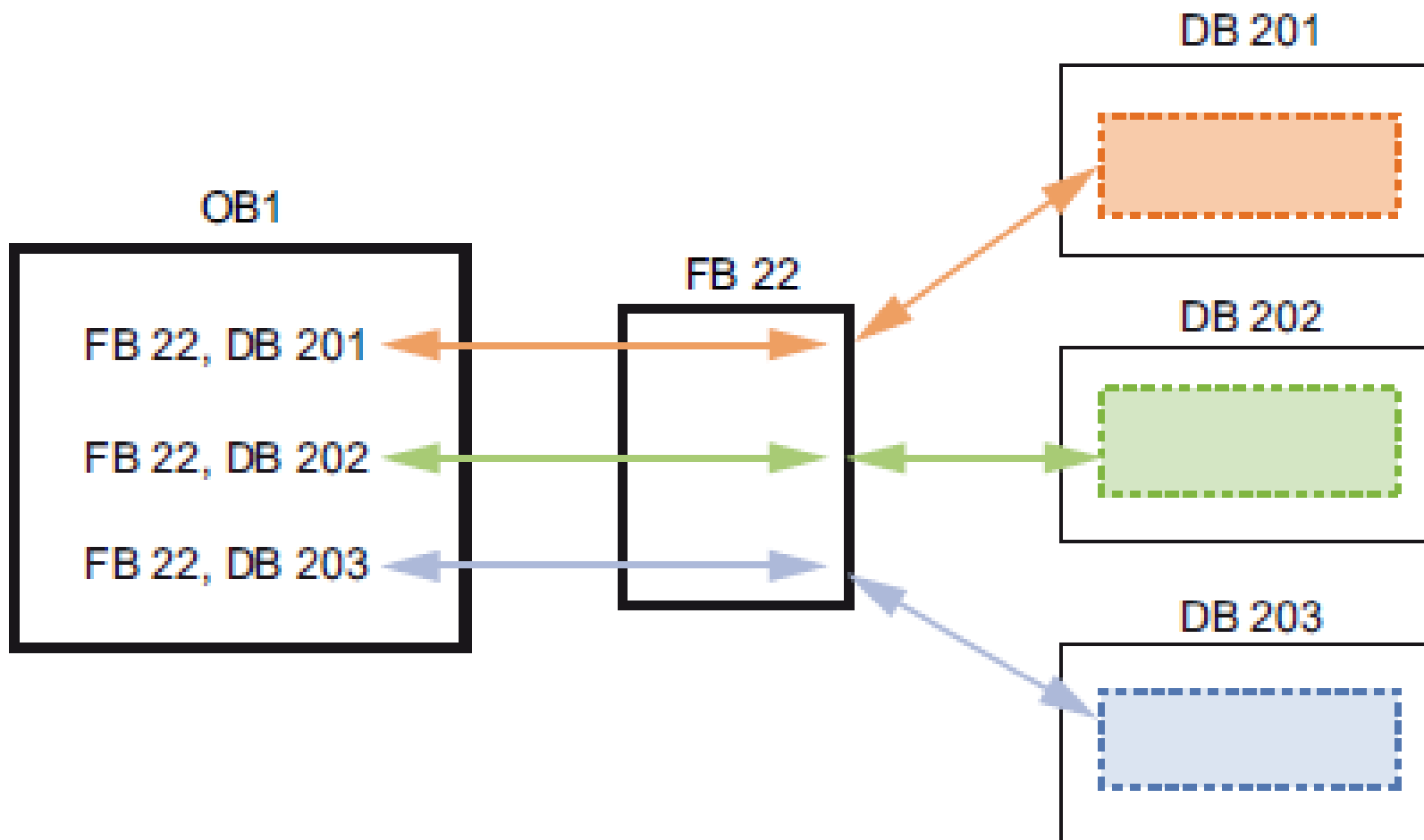
② – инструкция или събитие, което инициира изпълнението на друг блок

③ – изпълнение на програмата

④ – край на извикания блок (връщане към извикващия блок)



Модулна структура – дълбочина на вложени блокове



Изпълнение на един и същ функционален блок с различни блокове данни

Повикващият блок предава параметри на FB и също така идентифицира специфичен блок данни (DB), който съхранява данните за конкретното повикване или заявка от този FB. Промяната на зададената база данни позволява на FB с общо приложение да управлява работата на набор от устройства.

Тази структура позволява на един общ FB да контролира няколко подобни устройства, като например двигатели, чрез присвояване на различен блок данни за всяко повикване за различните устройства.

Всяка DB база данни съхранява данните (като скорост, време за ускоряване и общо време на работа) за отделно устройство.

В този пример FB 22 управлява три отделни устройства, като DB 201 съхранява оперативната база от данни за първото устройство, DB 202, съхранява оперативните данни за второто устройство, и DB 203 съхранява оперативните данни за третото устройство.

Обикновено FB се използват за управление на задачи или устройства, които не завършват действието си в рамките на един цикъл на сканиране. За съхраняване на работните параметри, така че да могат да бъдат бързо достъпни от едно сканиране към следващо, всеки FB в потребителската програма има една или повече оперативни бази данни.

Основни групи инструкции

Размерът на потребителската програма, данните и конфигурацията са ограничени от наличната памет за зареждане и работната памет в CPU. Няма конкретно ограничение за броя на отделните блокове OB, FC, FB и DB. Въпреки това, общият брой на блоковете е ограничен до 1024.

1. Инструкции от контактен тип (контактни инструкции). Това са инструкции за обработка на отделен бит. С тях се обработва т.нар. „контактни” битове от данновата област, съответстващи на логически елементи с две състояния – включено и изключено. Такива са инструкциите за двоичните логически операции AND, OR, XOR и др. Към този тип инструкции спадат и инструкциите, чрез които се кодират операциите за логическо установяване Set и нулиране Reset на бит от данновата област.

2. Инструкции за работа с числови операнди, представени чрез байт, дума, дълга дума и числа с плаваща запетая. Делят се на три вида:

– математически инструкции, чрез които се извършват математически операции върху операндите;

– логически инструкции, към които се причисляват операциите по преместване и ротиране на операндите;

– инструкции за преобразуване на типа на информацията, като например преобразуване от двоичен в двоично-десетичен вид и т.н.

3. Инструкции за сравняване, които сравняват операндите и изработват като изход еднобитов резултат. Резултатът от сравнението може да се използва като бит в изразите на контактните инструкции.

4. Инструкции за управление на хода на потребителската програма като условни разклонения, прескачане на отделни части от програмата, предаване на управлението, извикване на подпрограми и др.

5. Инструкции за ползване на системните функции, имитиращи работата на хардуерни устройства като таймери, броячи и др. Към тях се отнасят и инструкциите за управление на специализирани хардуерни блокове и др.

Ладер диаграми (Ladder logic - LAD)

Ладер диаграмата представлява графично описание на логически инструкции, в които булевите променливи са изразени чрез „контакти” и практически повтаря електрическата схема на изгражданите на базата на релейно-контактни схеми устройства. Състоянията на отделните битове се определя от две основни инструкции – „изпълни при затваряне” (examine if close –| |–) и „изпълни при отваряне” (examine if open –|/|–). Тези две инструкции напълно съответстват на релейните схеми „нормално отворен контакт” и „нормално затворен контакт”. Всъщност те изразяват по какъв начин участват булевите променливи в логическото уравнение – с правата или с инверсната си стойност.

По аналогия с релейно-контактните схеми, всяко логическо уравнение се представя чрез последователни или паралелни връзки между логическите променливи („контакти”) и формира т.нар. „стъпало” (contact network) в ладер диаграмата. Началото на всяко стъпало започва с условна „захранваща линия” (power line). Резултантното състояние на всеки изходен (или междинен) бит се представя чрез „бобина“ (coil – ()), чийто логически смисъл е присвояване на стойност.

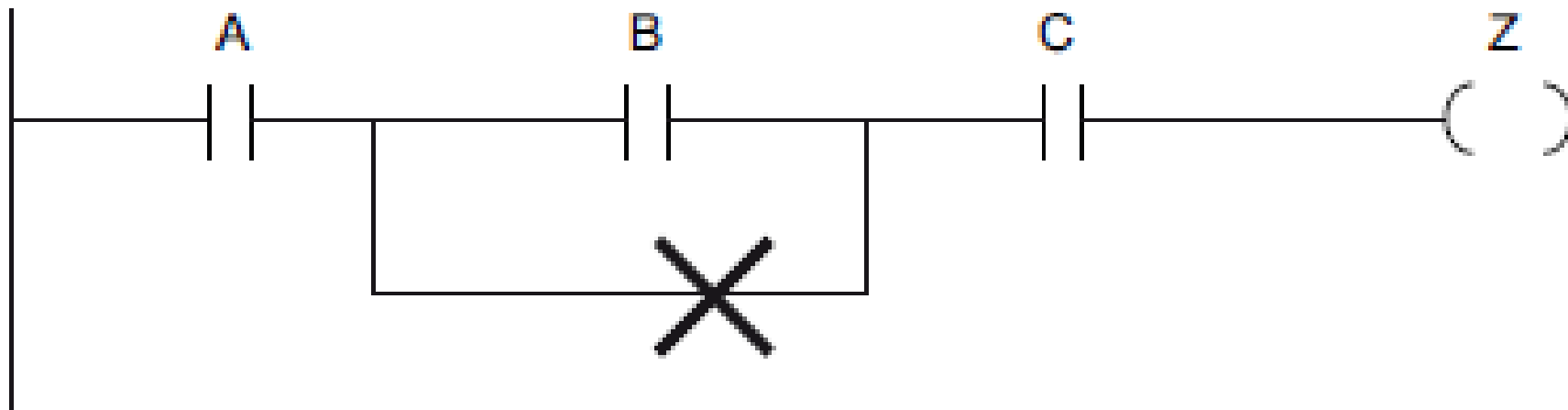
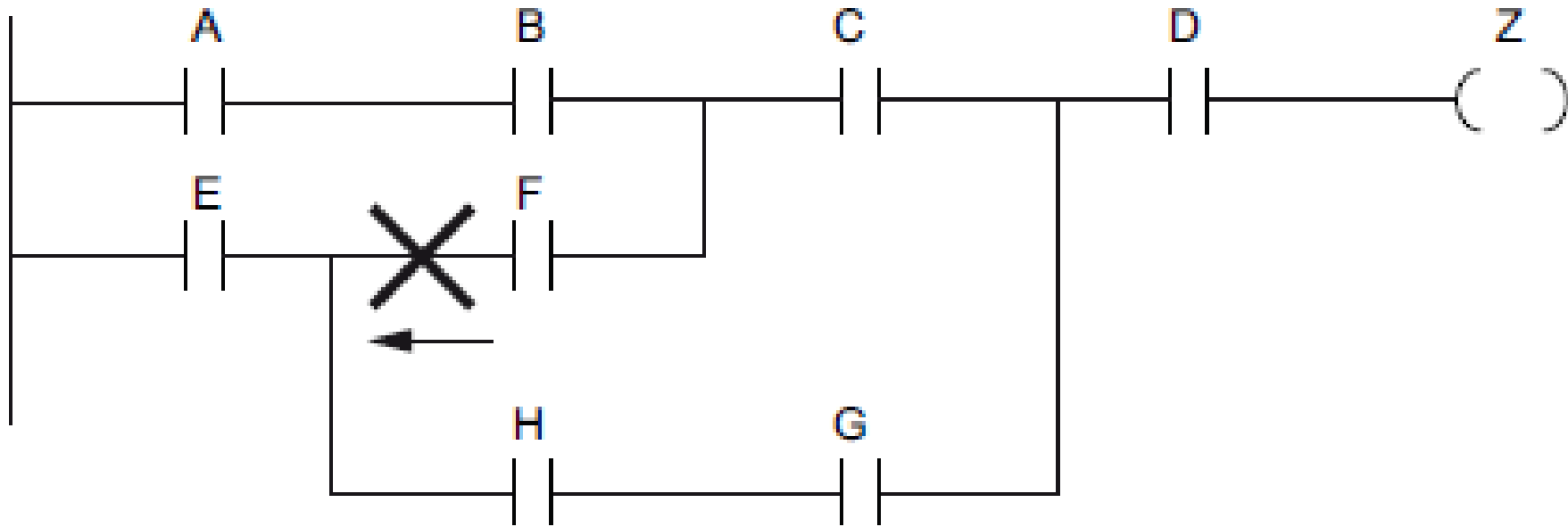
Ладер диаграми (Ladder logic - LAD)

Отделните стъпала в ладер диаграмите се обработват от операционната система на промишления контролер последователно отгоре надолу. Логическата комбинация между отделните контакти в ладер диаграмите се нарича „тестова зона“ или „логически условия“, а участващите контактни инструкции – „входни инструкции“.

Резултатът от логическата операция може да служи като условие за извършването на определена логическа операция. За тези цели се ползват т.нар. „специални бобини“ (special coil – (xxx)). Те могат да бъдат:

1. Функции на схемите:
 - установяване на бит в 1 – Set – (S) ;
 - нулиране на бит – Reset – (R) .
2. Функции за организиране на програмата:
 - безусловен преход към етикет – Jump to Label – (J) ;
 - извикване на подпрограма с номер n – Call Subroutine – (Call n);
 - връщане от подпрограма – Return from Subroutine – (Ret) ;
 - край на потребителска програма – End of Program – (END) .

Ладер диаграми (Ladder logic - LAD)



Ладер диаграми (Ladder logic - LAD)

Условната „бобина“ се нарича още „изходна инструкция“, което означава, че в зависимост от резултата на изпълнението на логическото уравнение могат да се изпълняват значително по-сложни инструкции, като например „бокс инструкции“ (box).

Бокс инструкциите съчетават действието на контактните и числовите инструкции. Те се разделят на две основни групи:

- бокс изходни инструкции за системни функции (system function box), които са вградени в операционната система, като таймери, броячи и др.

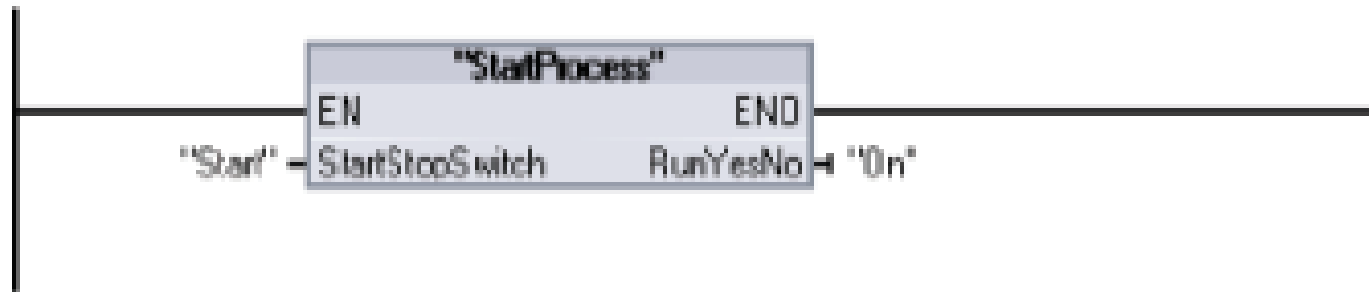
- бокс изходни функции за операционни символи (operational box), които представляват логически и математически операции над числови операнди, както и операции за прехвърляне на променливи в различните даннови области.

Нормално бокс инструкциите могат да се използват само като изходни инструкции. Изключение е бокс инструкцията за сравнение (comparison box instruction), която извършва операция над числови операнди и формира двоичен (булев) резултат

Някои инструкции (напр. математически или за преместване) предоставят параметри за разрешение (EN и ENO). Тези параметри се отнасят до подаването на захранване в LAD или FBD и определят дали инструкцията ще се изпълнява по време на това сканиране.

- EN (Enable In) е двоичен вход. На този вход трябва да има захранване (EN = 1), за да се изпълни бокс инструкцията. Ако EN входът на LAD бокс е свързан директно към захранването, инструкцията винаги ще бъде изпълнена.

- ENO (Enable Out) е двоичен изход. Ако боксът получава захранване на входа EN и изпълнява функцията си без грешка, тогава изходът на ENO подава захранване (ENO = 1) на следващия елемент. Ако се установи грешка при изпълнението на инструкцията, тогава захранването се прекратява (ENO = 0) в бокс инструкцията, която генерира грешката.



- Нормално отвореният контакт е затворен (ON), когато стойността на зададения бит е равна на 1.
- Нормално затвореният контакт е затворен (ON), когато стойността на зададения бит е равна на 0.
- Контакти, свързани последователно реализират И логическа функция.
- Свързаните в паралел контакти създават ИЛИ логическа функция.

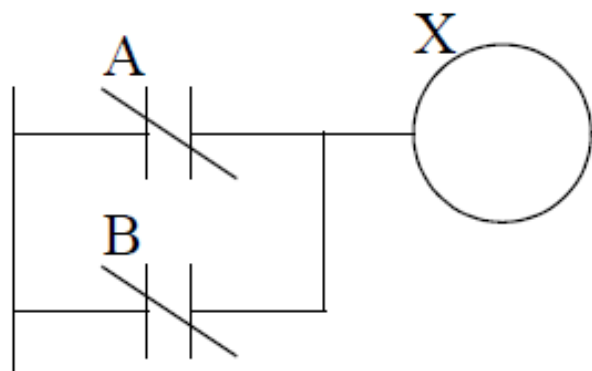


Parameter	Data type	Description
IN	Bool	Assigned bit

NAND

$$X = \overline{A \cdot B}$$

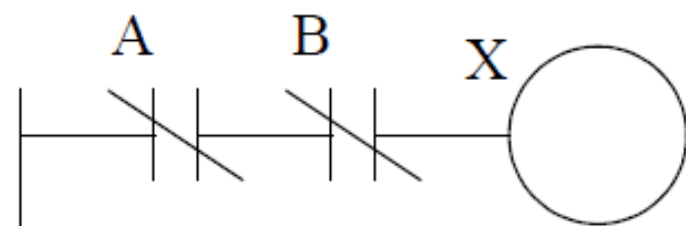
$$X = \bar{A} + \bar{B}$$



NOR

$$X = \overline{A + B}$$

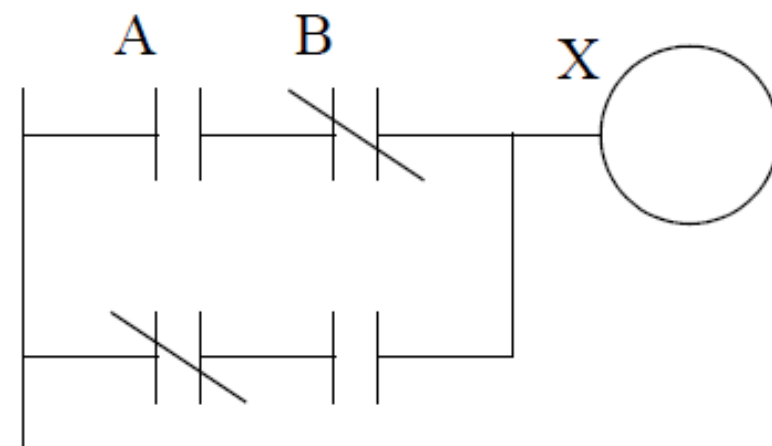
$$X = \bar{A} \cdot \bar{B}$$



EOR

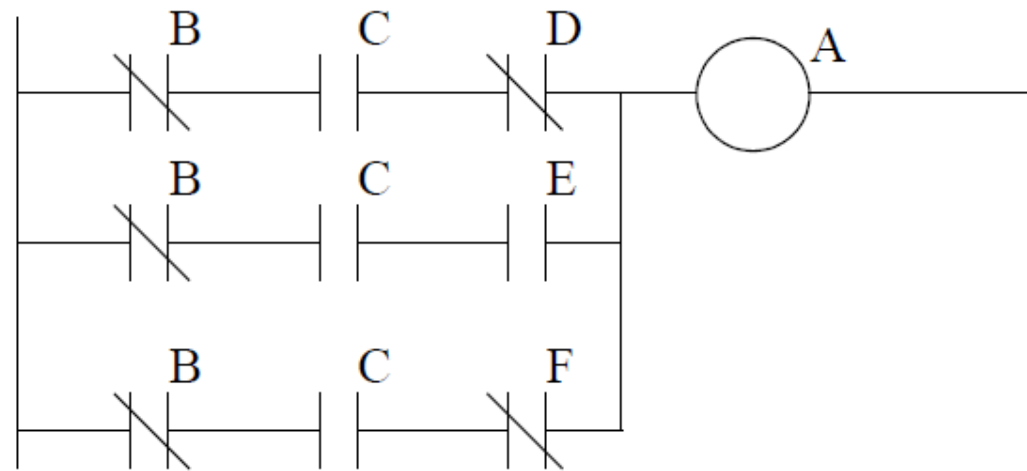
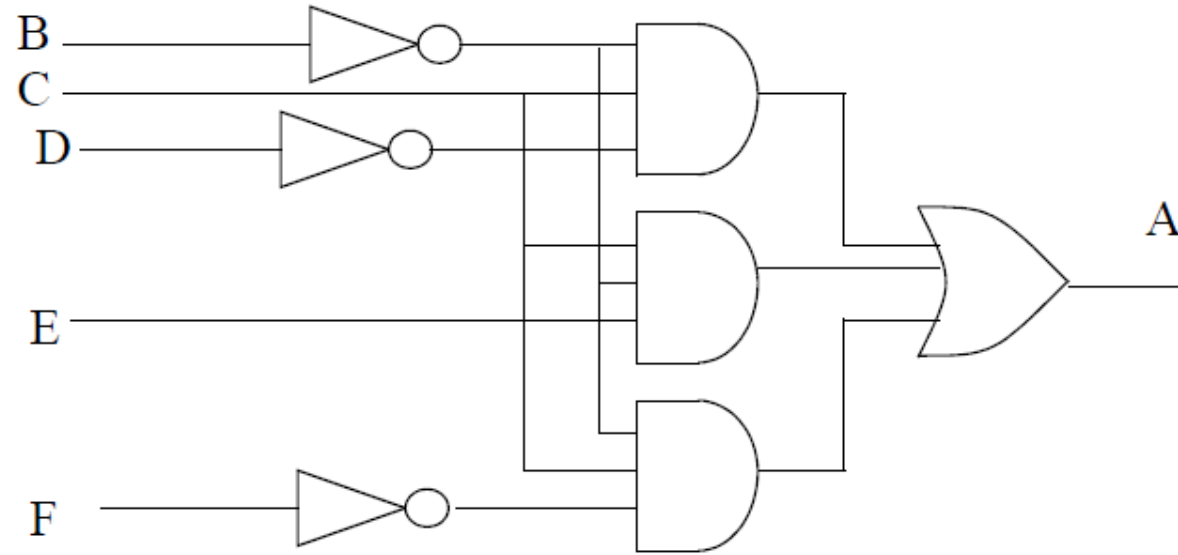
$$X = A \oplus B$$

$$X = A \cdot \bar{B} + \bar{A} \cdot B$$

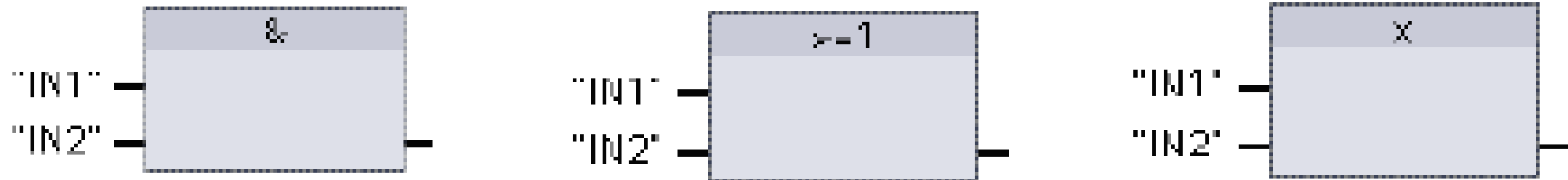


Програмируеми логически контролери

$$A = (\bar{B} \cdot C \cdot \bar{D}) + (\bar{B} \cdot C \cdot E) + (\bar{B} \cdot C \cdot \bar{F})$$



При програмирането с използването на FBD, LAD контактните стъпала се трансформират в AND (&), OR (≥ 1) и XOR (x) бокс инструкции, където може да се задават стойности на битовете за входовете и изходите. Може също така да се свързват с други логически бокс инструкции и да се създават собствена комбинационна логика.

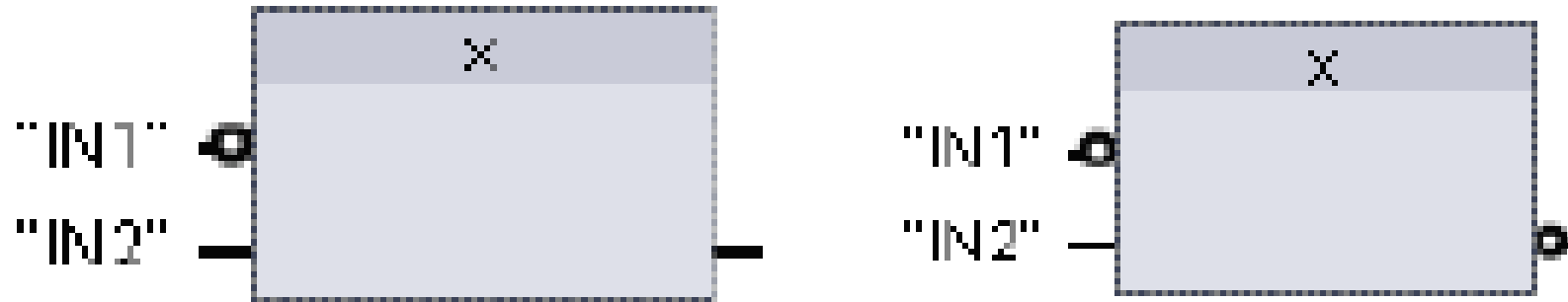


Parameter	Data type	Description
IN1, IN2	Bool	Input bit

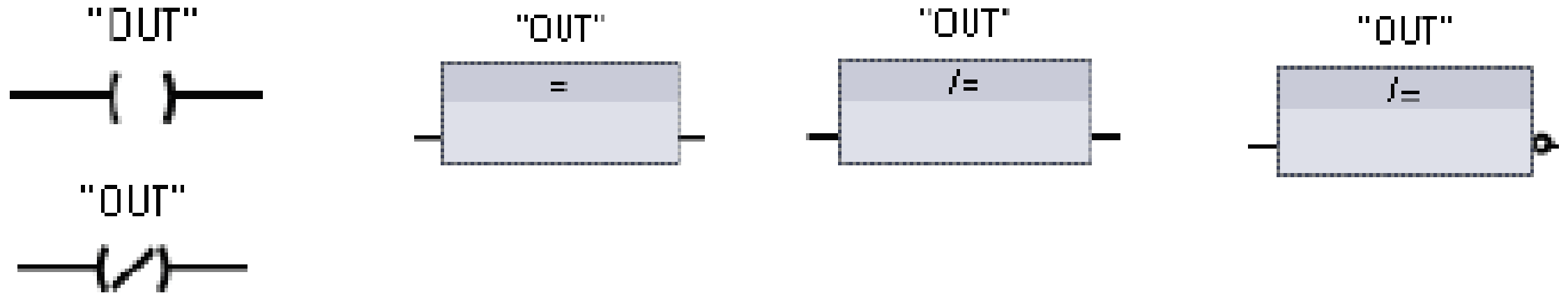
Контактът LAD NOT инвертира логическото състояние на входа.

1) Ако в контакта NOT не постъпва захранване, тогава се подава захранване навън.

2) Ако в NOT контакт постъпва захранване, тогава не се подава захранване навън.



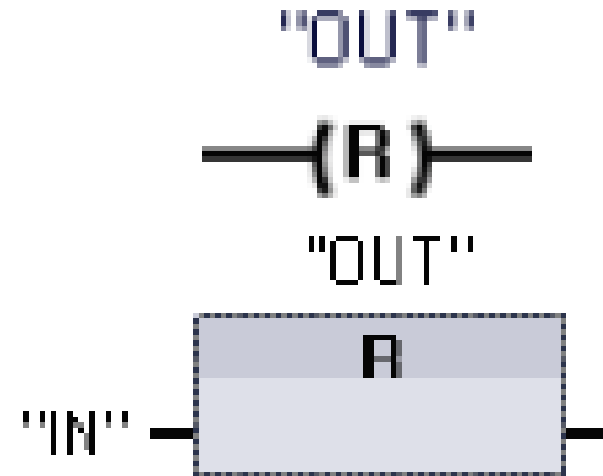
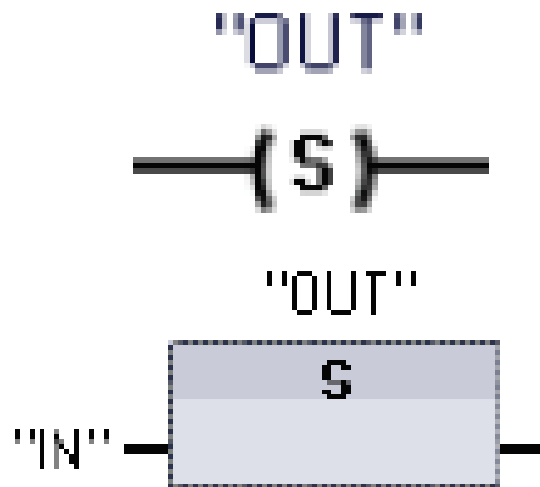
- Ако има захранване към изходна бобина или FBD "=" бокс инструкцията е разрешена, тогава изходният бит е равен на 1.
 - Ако няма захранване към изходна бобина или FBD "=" бокс инструкцията не е разрешена, тогава изходният бит е равен на 0.
 - Ако има захранване към инвертирана изходна бобина или FBD "/ =" бокс инструкцията е разрешена, тогава изходният бит е равен на 0.
 - Ако няма захранване към инвертирана изходна бобина или FBD "/ =" бокс инструкцията не е разрешена, тогава изходният бит е равен на 1.



Parameter	Data type	Description
OUT	Bool	Assigned bit

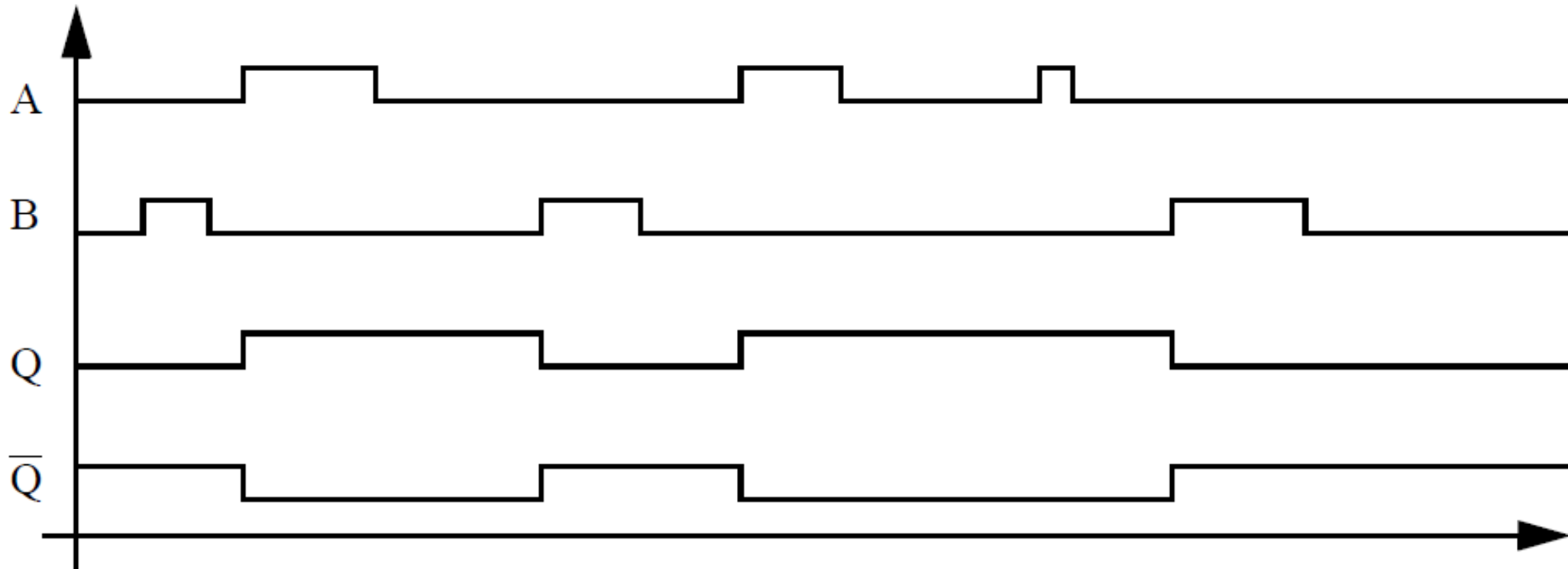
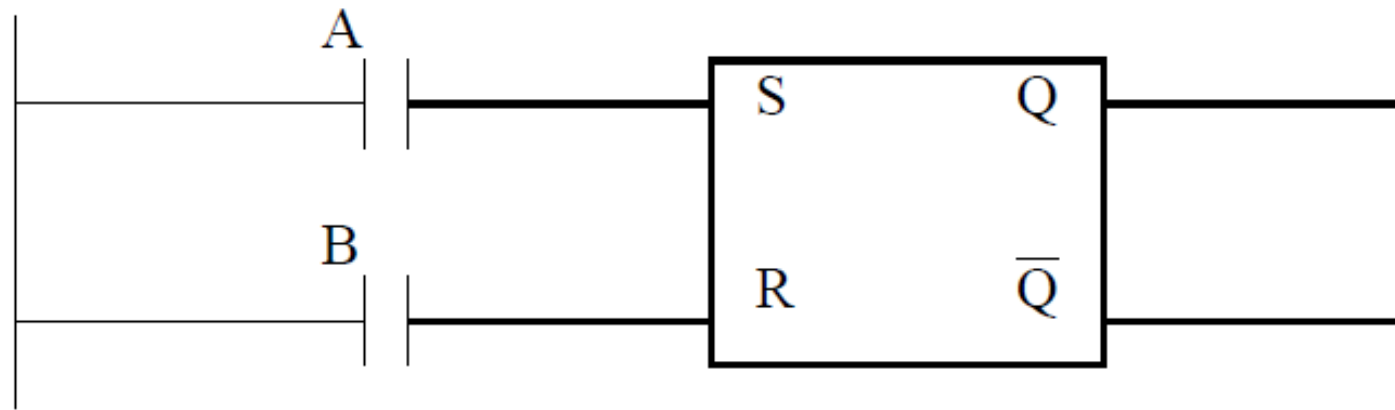
Когато S (Set) е активиран, стойността на данните в адреса OUT е равна на 1. Когато S не е активиран, стойността на OUT не се променя.

Когато се активира R (Reset), стойността на данните в адреса OUT е равна на 0. Когато R не е активиран, стойността на OUT не се променя.

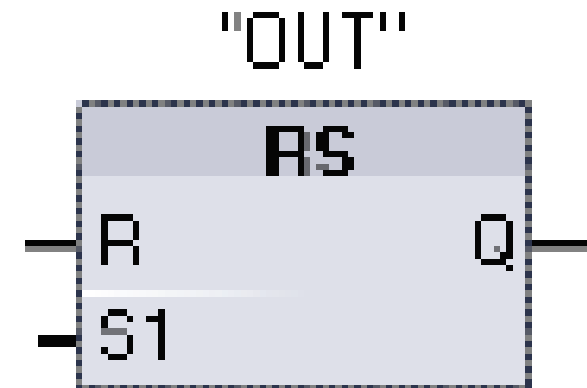
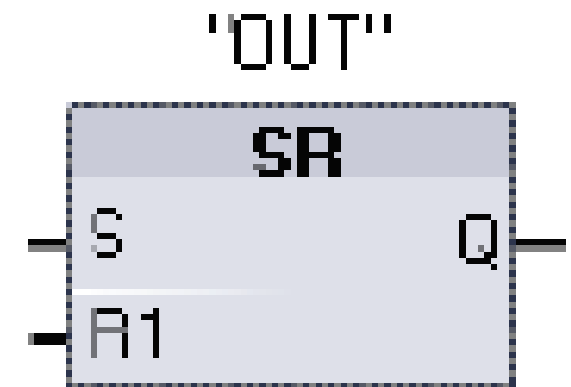


Parameter	Data type	Description
IN (or connect to contact/gate logic)	Bool	Bit location to be monitored
OUT	Bool	Bit location to be set or reset

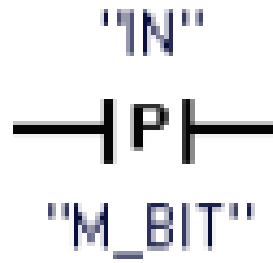
Програмируеми логически контролери



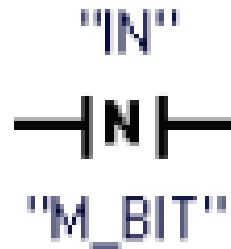
Instruction	S1	R	"OUT" bit
RS	0	0	Previous state
	0	1	0
	1	0	1
	1	1	1
SR	S	R1	
	0	0	Previous state
	0	1	0
	1	0	1
	1	1	0



Parameter	Data type	Description
S, S1	Bool	Set input; 1 indicates dominance
R, R1	Bool	Reset input; 1 indicates dominance
OUT	Bool	Assigned bit output "OUT"
Q	Bool	Follows state of "OUT" bit



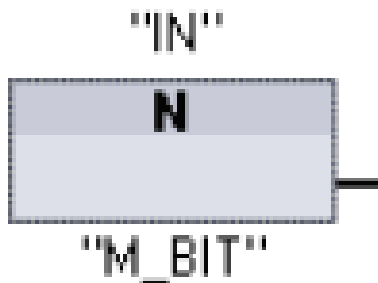
Състоянието на този контакт е TRUE при откриване на положителен преход (OFF-to-ON) на зададения бит "IN".



Състоянието на този контакт е TRUE при откриване на отрицателен преход (ON-to-OFF) на зададения бит "IN".



Изходното логическо състояние е TRUE при откриване на положителен преход (OFF-to-ON) на зададения входен бит.



Изходното логическо състояние е TRUE при откриване на отрицателен преход (ON-to-OFF) на зададения входен бит.



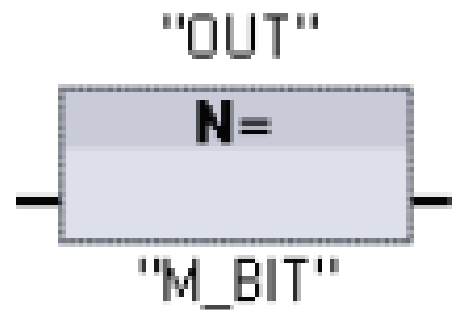
Зададеният бит "OUT" е TRUE, когато е открит положителен преход (OFF-to-ON) на захранването към бобината.



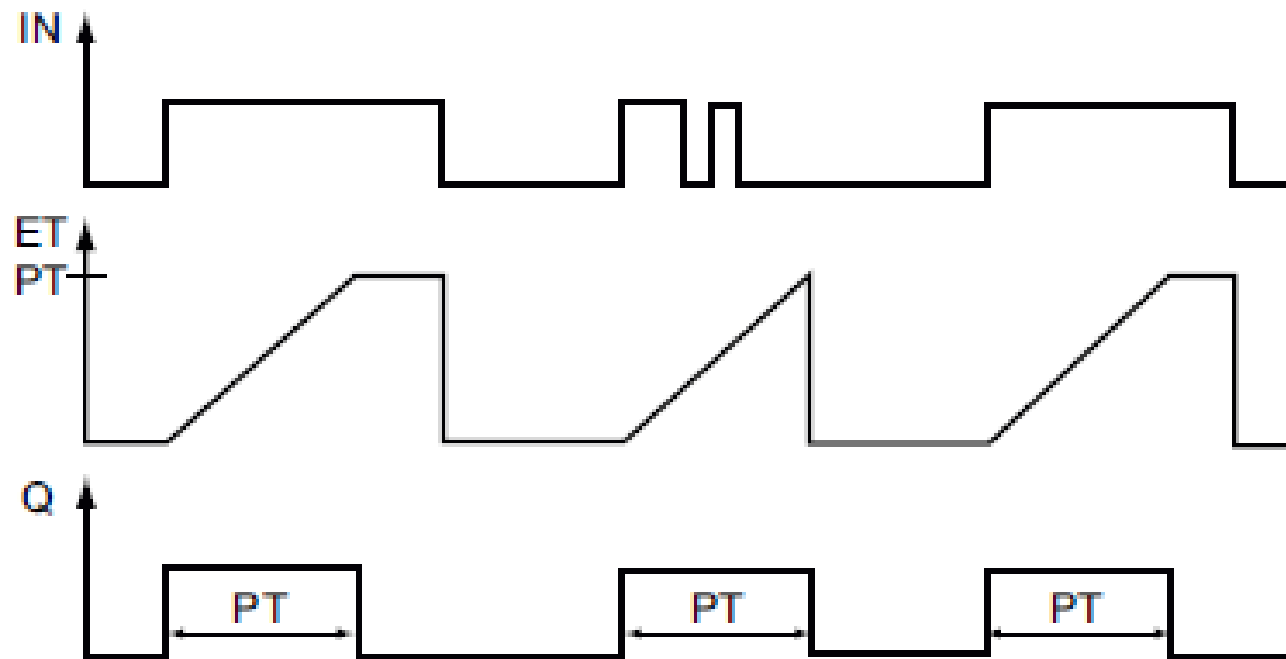
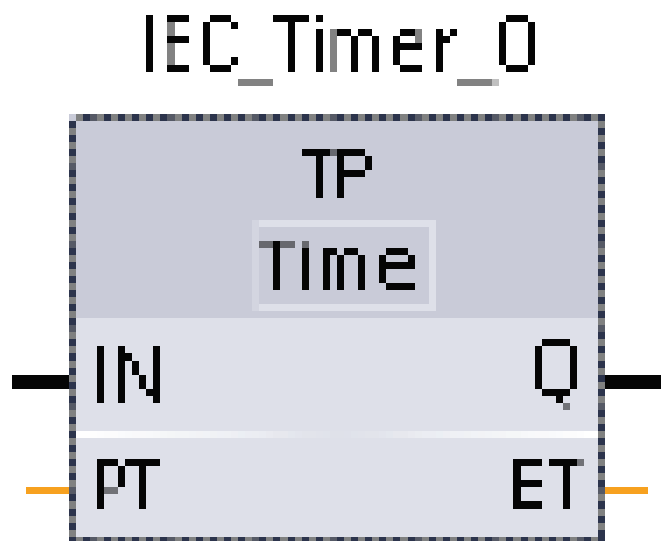
Зададеният бит "OUT" е TRUE, когато е открит отрицателен преход (ON-to-OFF) на захранването към бобината.



Зададеният бит "OUT" е TRUE, когато е открит положителен преход (OFF-to-ON) на входа на бокс инструкцията.

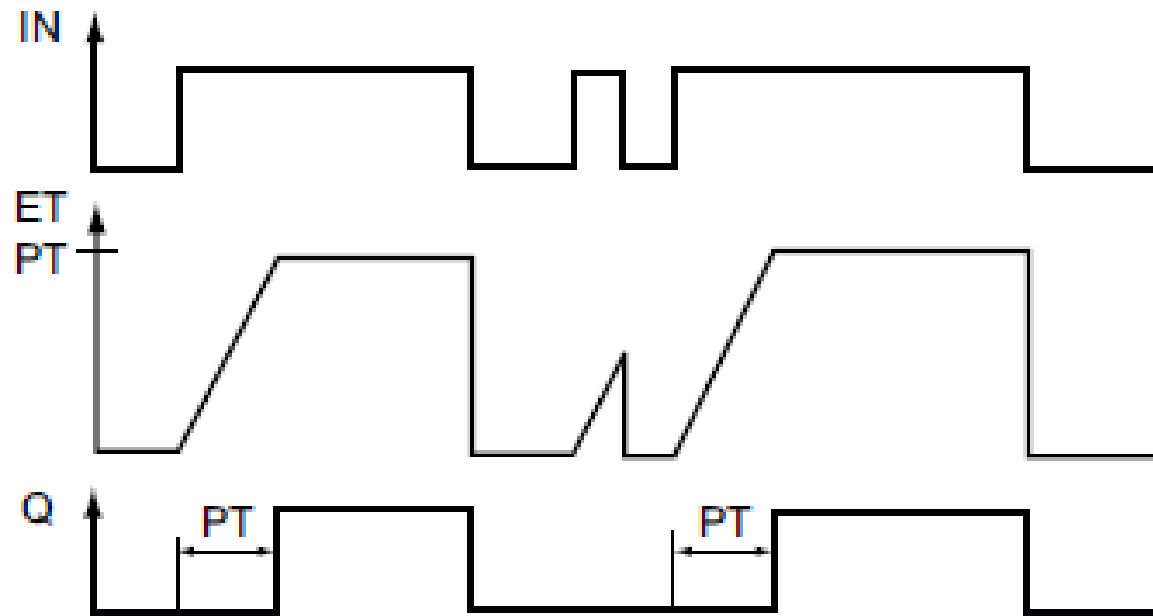


Зададеният бит "OUT" е TRUE, когато е открит отрицателен преход (ON-to-OFF) на входа на бокс инструкцията.



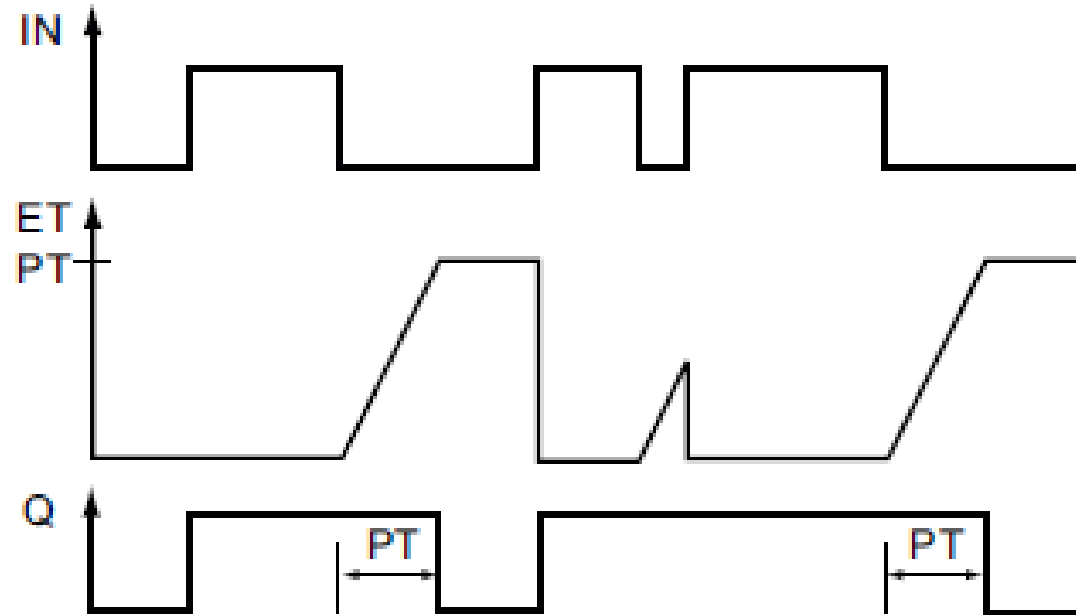
TP: Pulse timer

TP таймерът генерира импулс с предварително зададена продължителност.



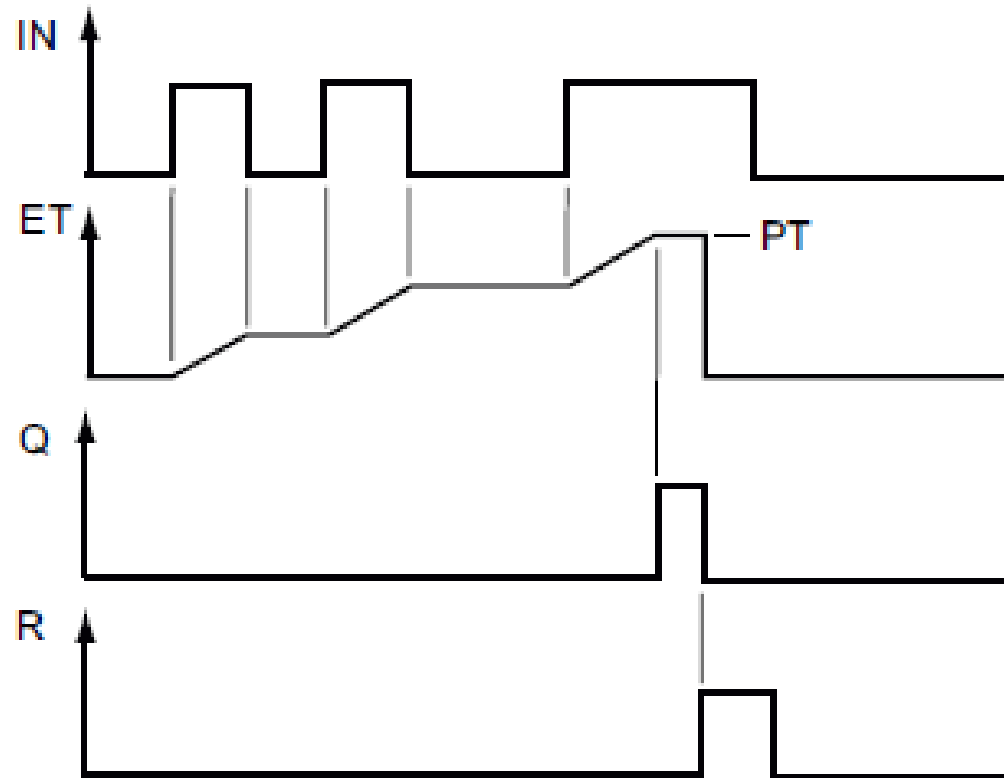
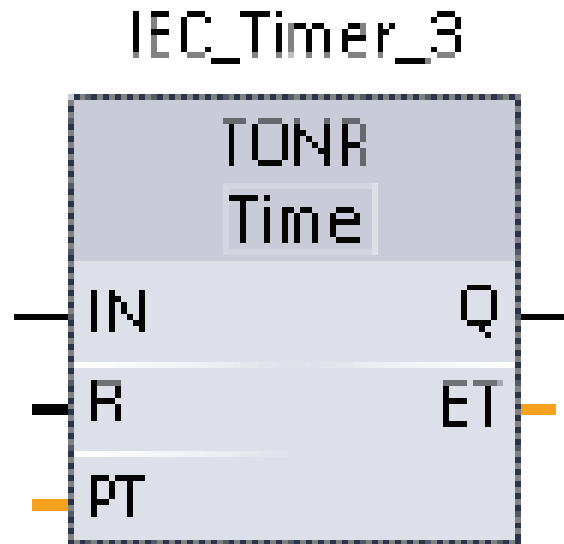
TON: ON-delay timer

TON таймерът установява изхода Q в състояние ON след предварително зададено закъснение.



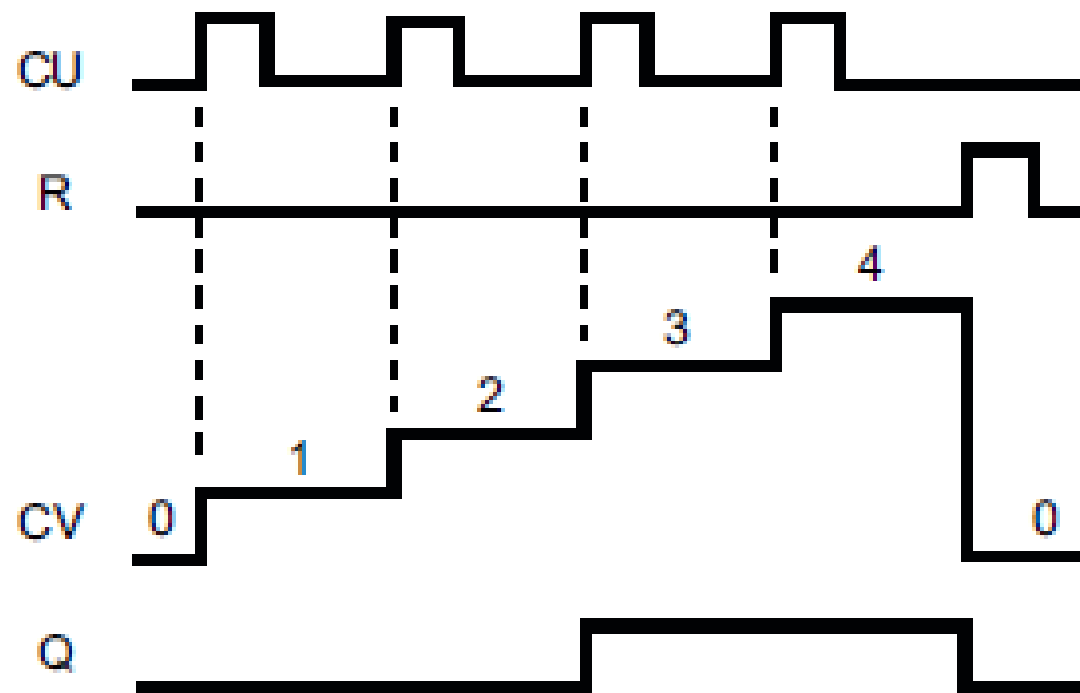
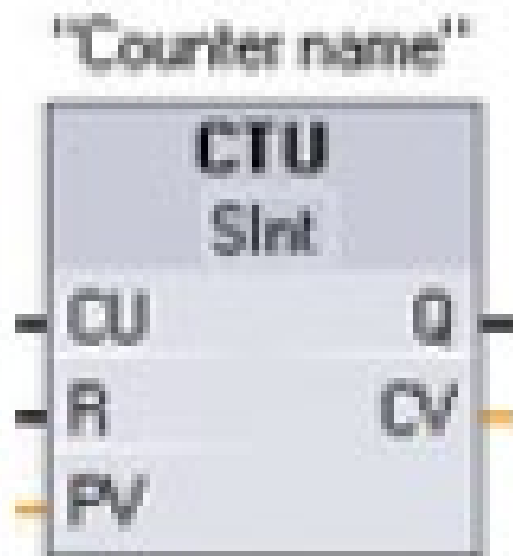
TOF: OFF-delay timer

TOF таймерът нулира изхода Q след предварително зададено закъснение.

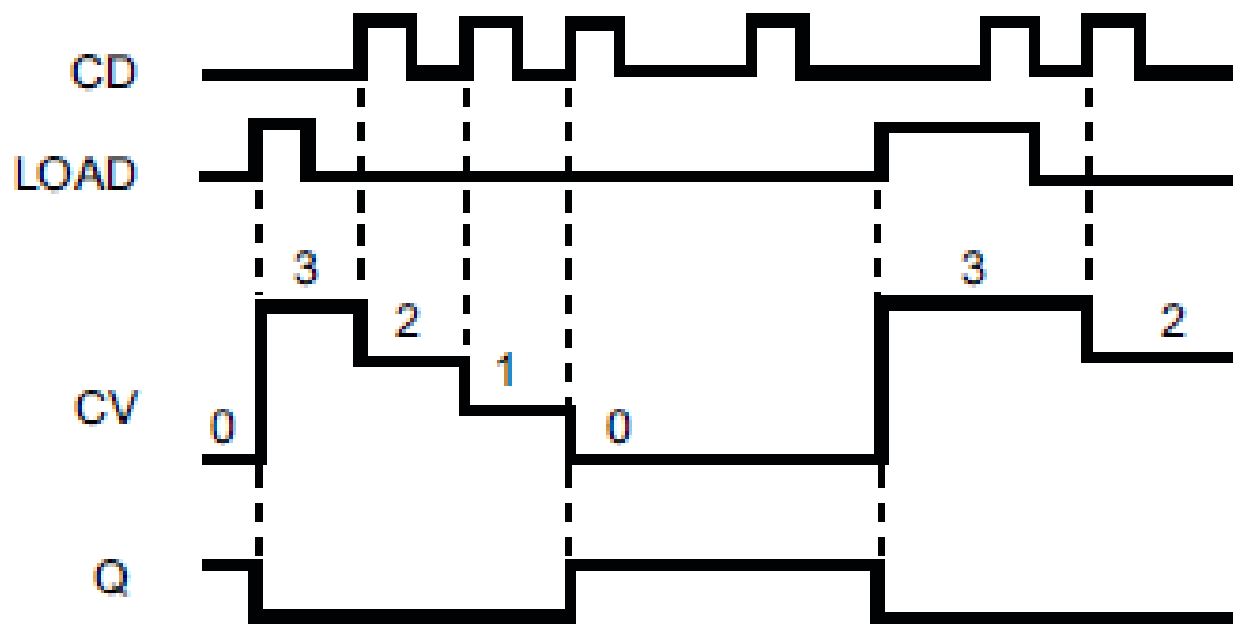
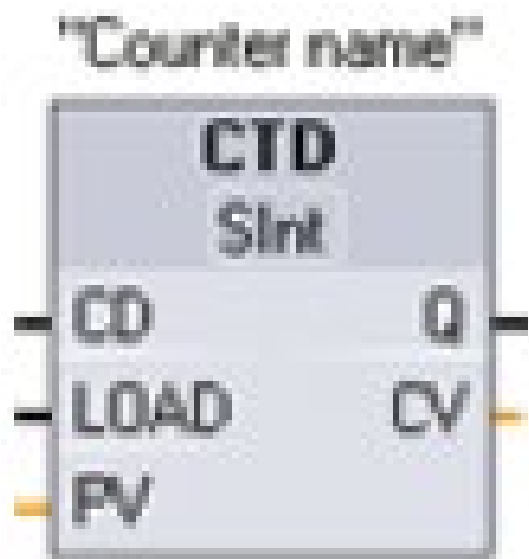


TONR: ON-delay Retentive timer

TONR таймерът установява изхода Q в ON след предварително зададено закъснение. Изтеклото време се акумулира в множество времеви периоди докато входа R не бъде използван за нулиране на изтеклото време.

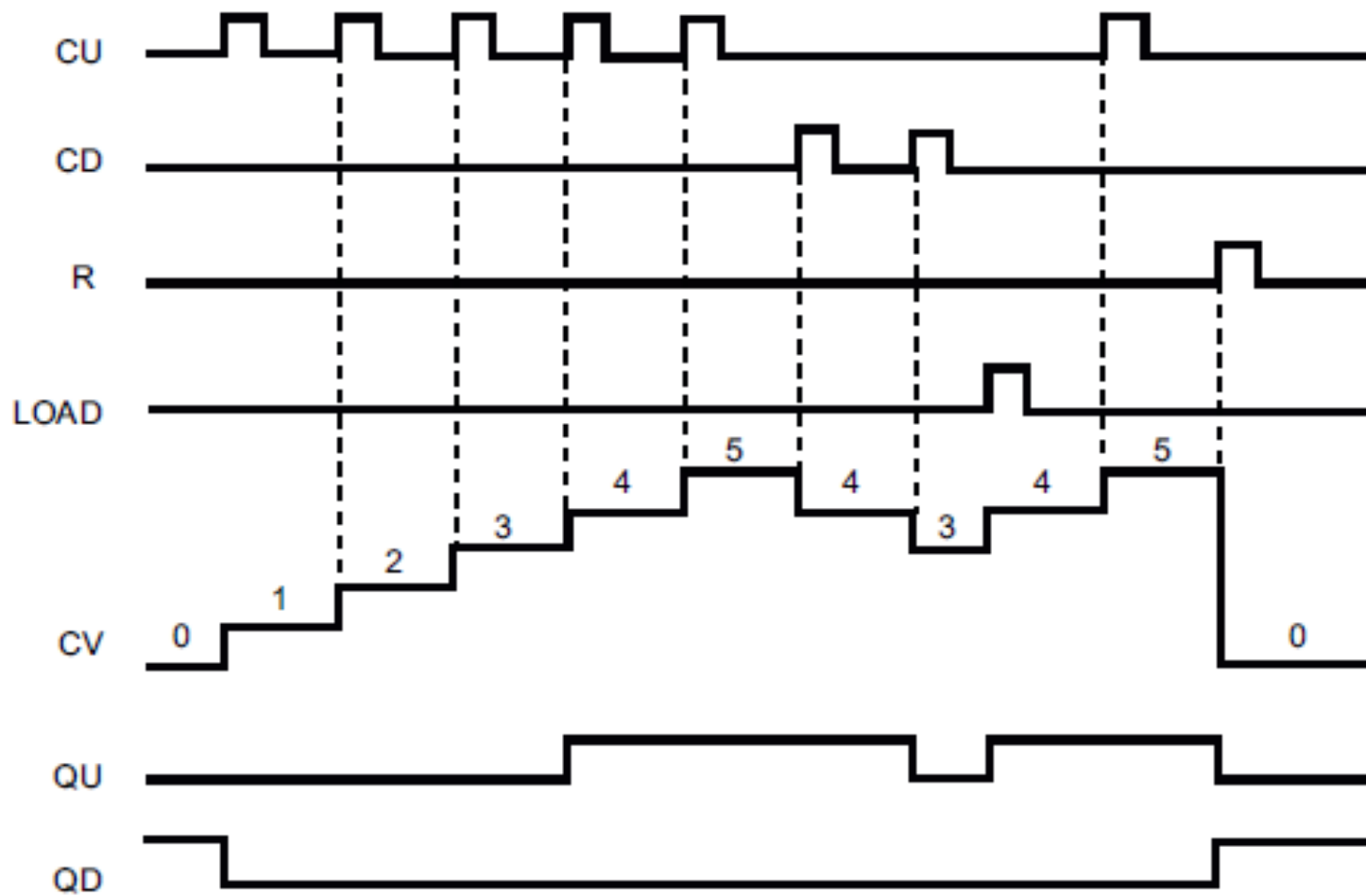
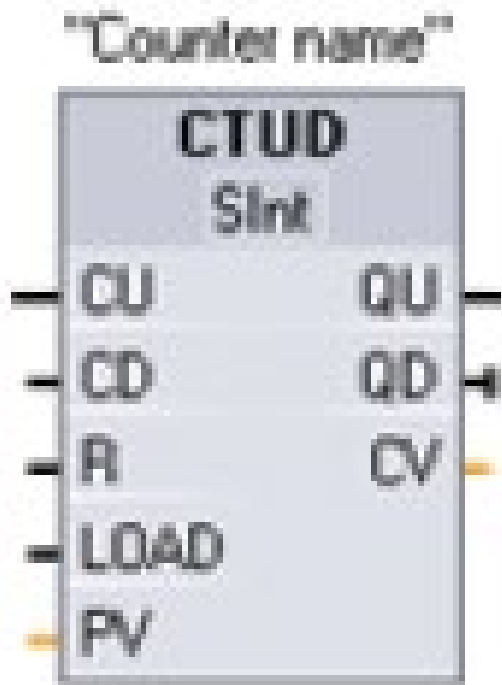


Сумиращ брояч



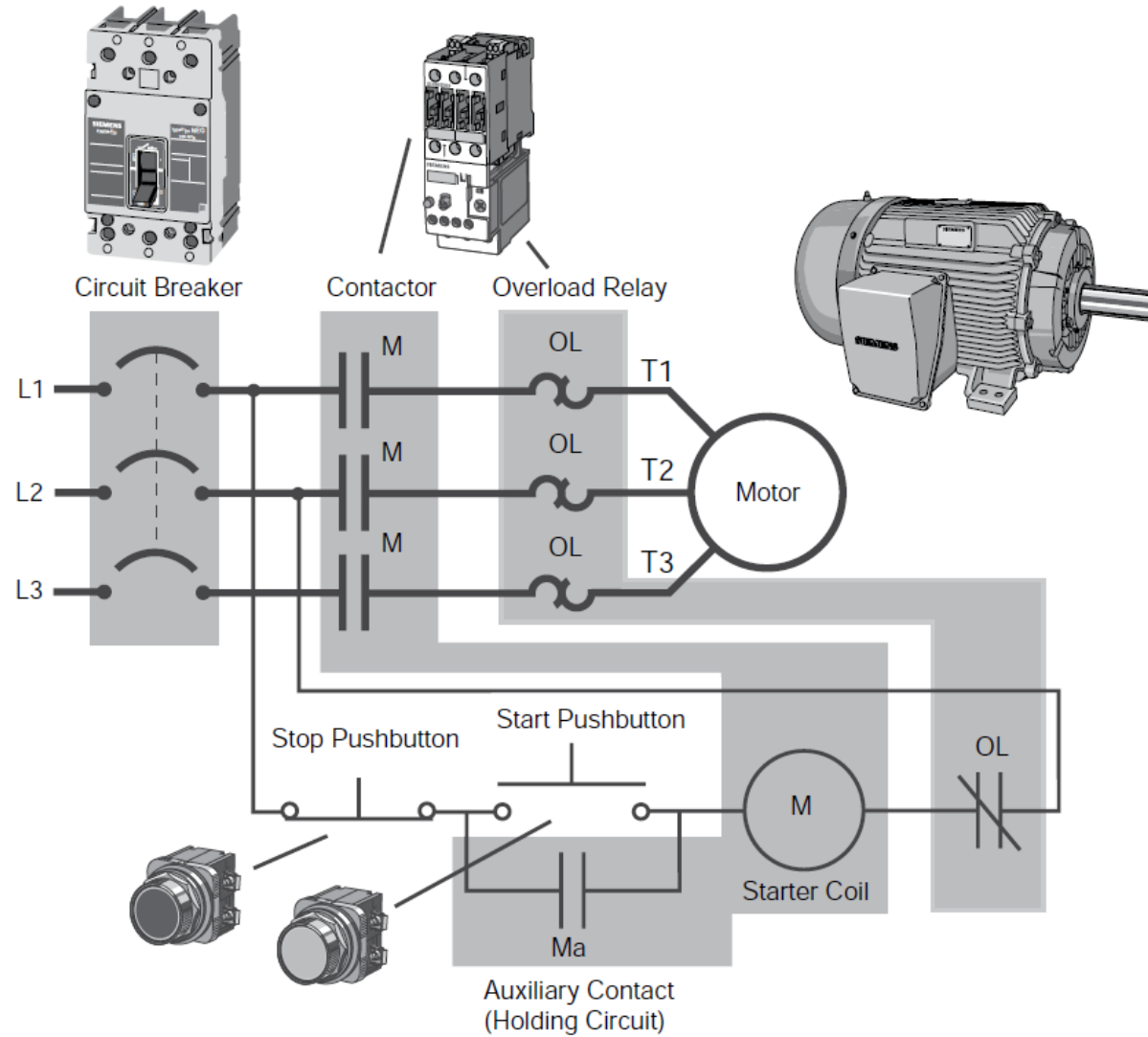
Изваждащ брояч

Програмируеми логически контролери

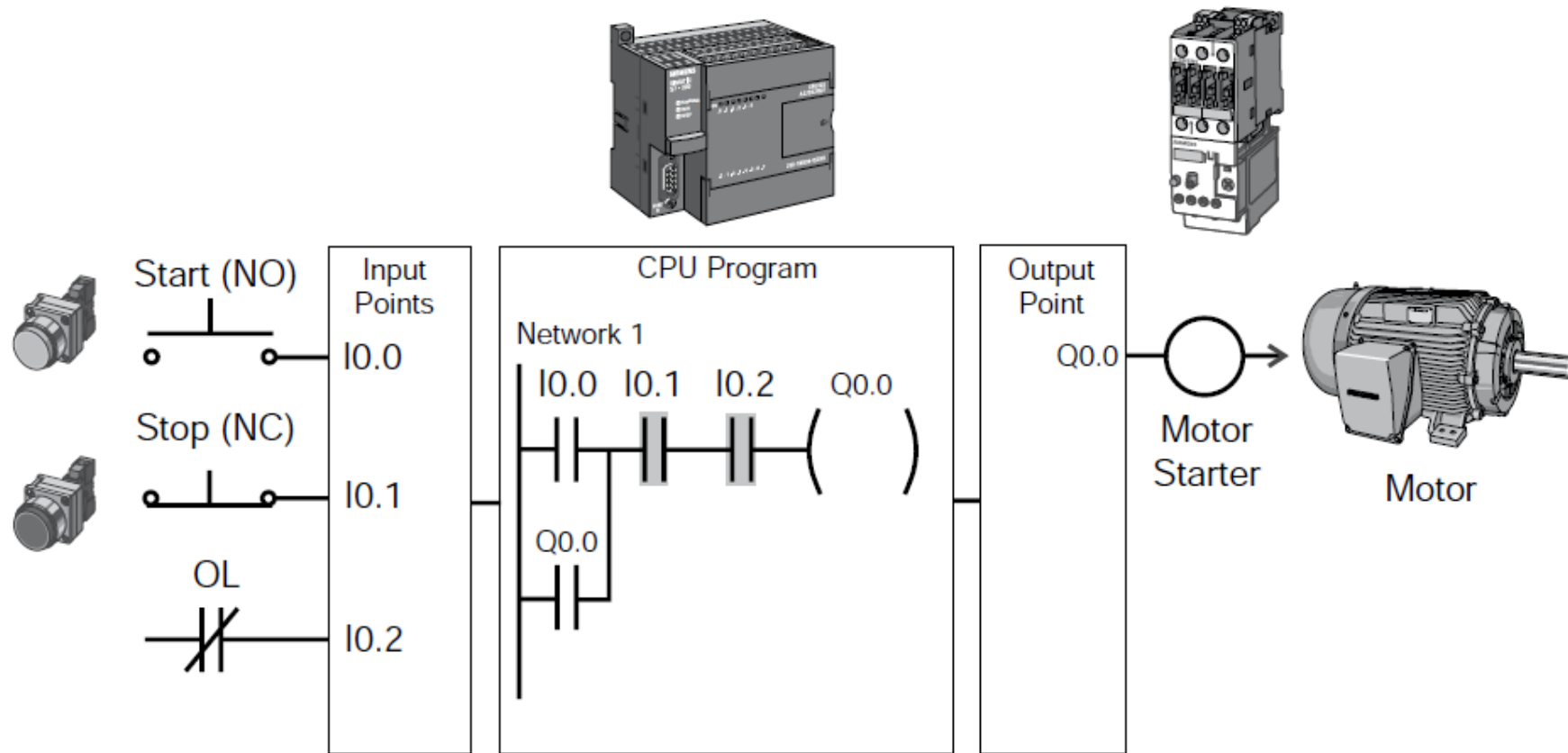


Реверсивен брояч

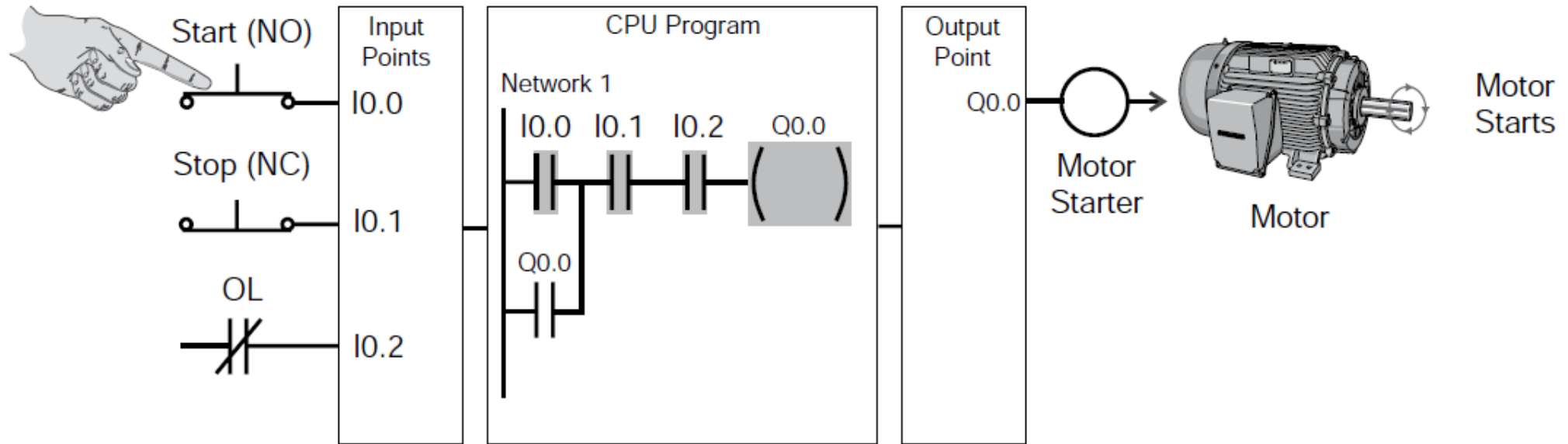
Реализиране на старт-стоп



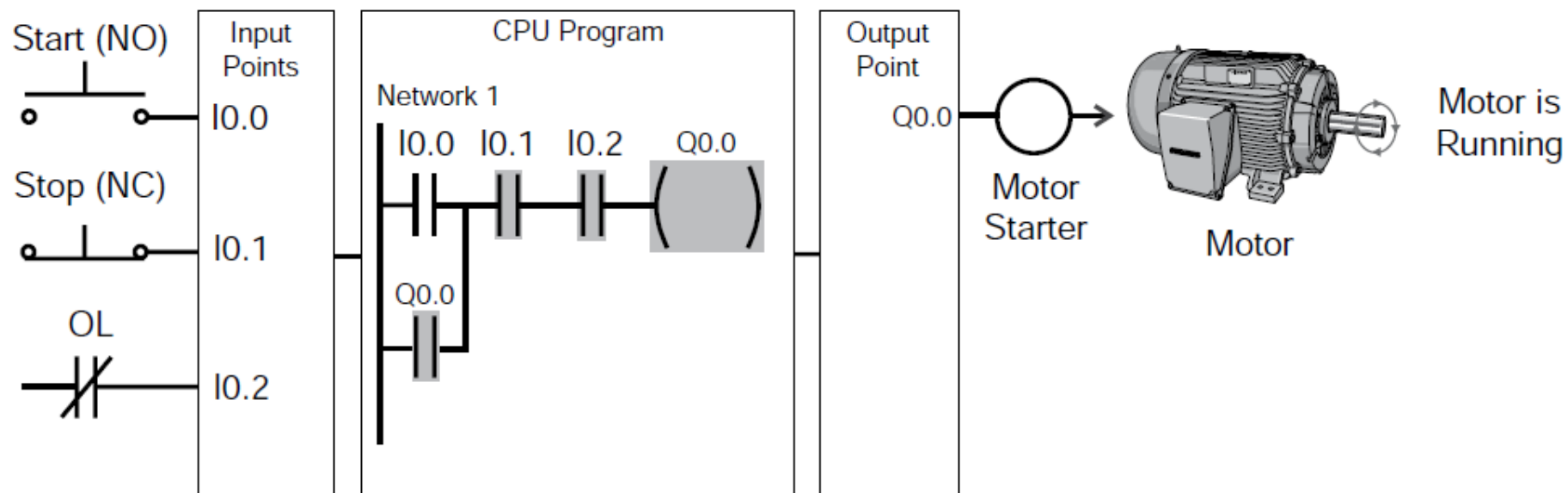
Реализиране на старт-стоп



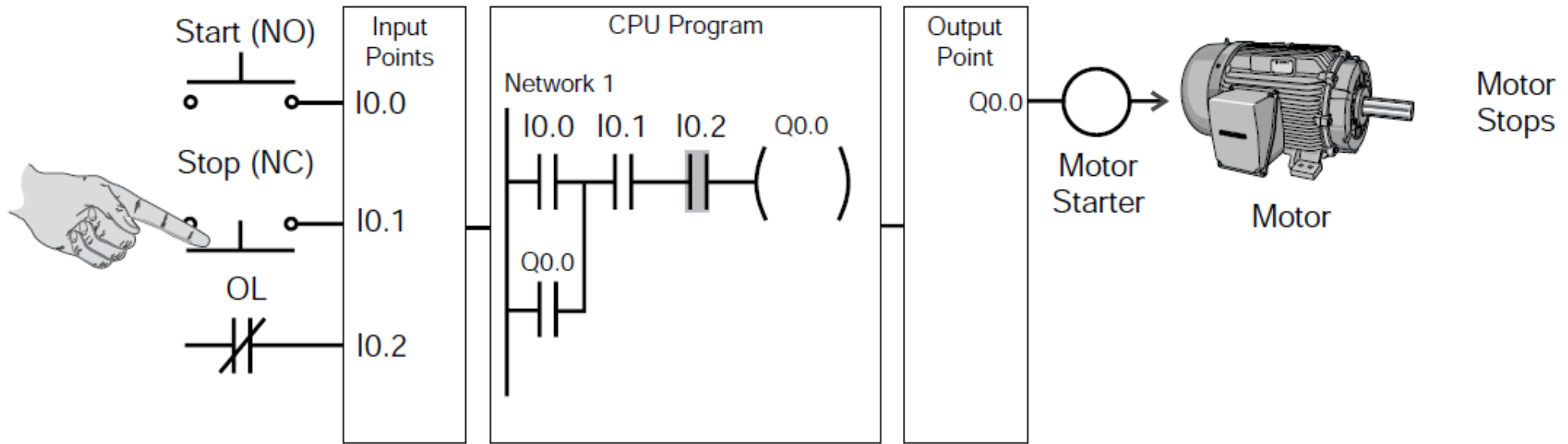
Реализиране на старт-стоп



Реализиране на старт-стоп

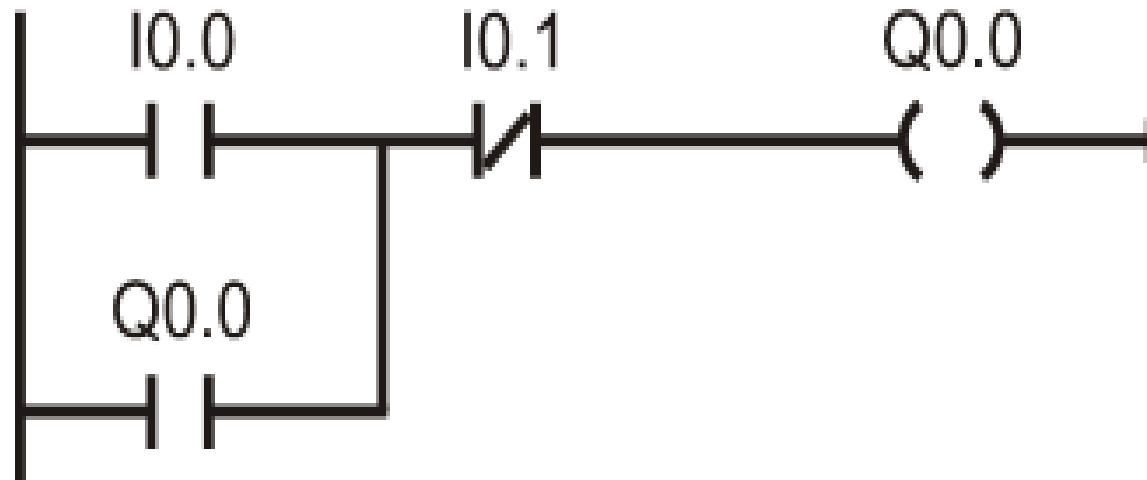


Реализиране на старт-стоп



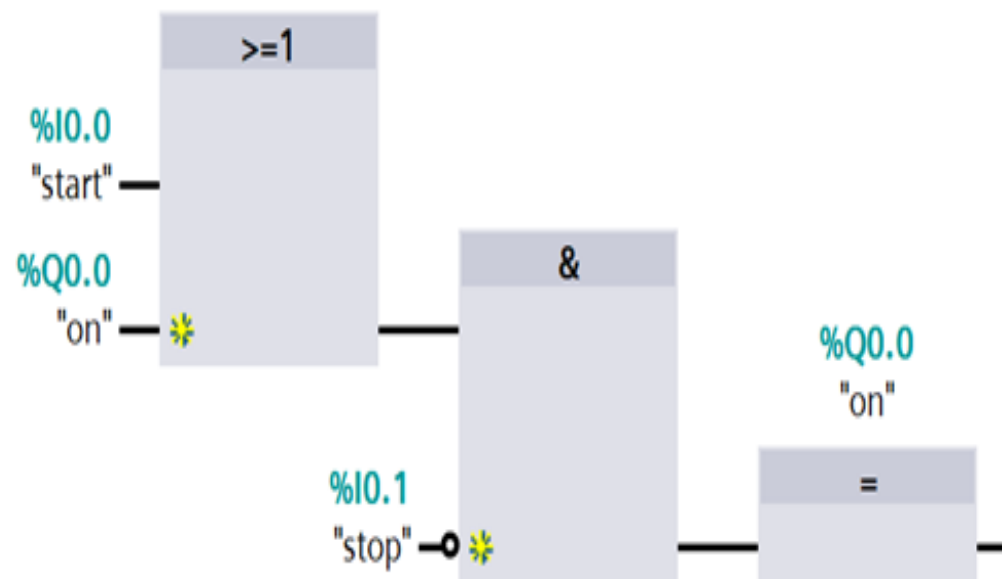
Реализиране на старт-стоп

Ладер диаграми (Ladder logic - LAD)



Реализиране на старт-стоп

Функционални блокови схеми (Function Block Diagram - FBD)



Реализиране на старт-стоп

SCL (structured control language) – програмен език от високо ниво, базиран на PASCAL

“On” := (“Start” OR “On”) AND NOT “Stop”;