Паралелни интерфейси

Автор: гл. ас. д-р инж. Любомир Богданов

Европейски съюз

ПРОЕКТ BG051PO001--4.3.04-0042
„Организационна и технологична инфраструктура за учене през целия живот и развитие на компетенции”
Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз
Инвестира във вашето бъдеще!

ЕСФ България
Европейски социален фонд

Съдържание

Паралелни асинхронни интерфейси

Различните видове интерфейси могат да бъдат разделени по следните признаци:

* в зависимост от типа на използваните сигнали биват **цифрови** и **аналогови**.
* в зависимост от използваната величина за предаване на информация биват **напрежителни** и **токови**.
* в зависимост от полярността на сигнала биват **еднополярни** и **двуполярни**.
* в зависимост от типа на трансивъра биват **несиметрични** и **диференциални**.
* в зависимост от това дали се използва тактов сигнал или не биват **синхронни** и **асинхронни**.
* в зависимост от начина на предаване на информацията биват **серийни** и **паралелни**.

Паралелни асинхронни/синхронни интерфейси

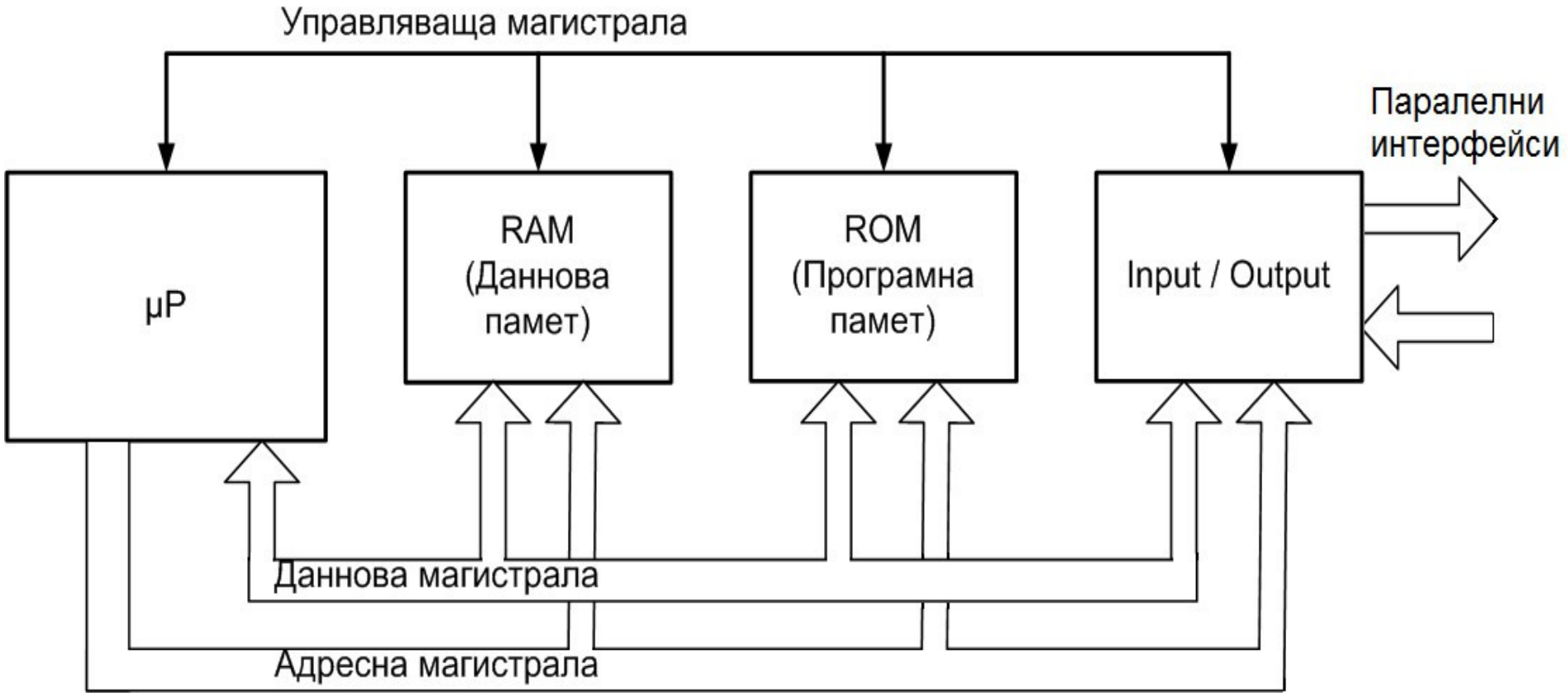
При паралелните интерфейси всеки бит от предаваните данни си има свой проводник, по който се предава/приема.

Това обуславя големия брой проводници, използвани при този тип интерфейси.

Използват се най-често за обмен на информация между микропроцесора и вътрешни периферни устройства.

В структурната схема на една микропроцесорна система адресните, данновите и управляващите са реализирани с паралелни интерфейси.

Паралелни асинхронни/синхронни интерфейси



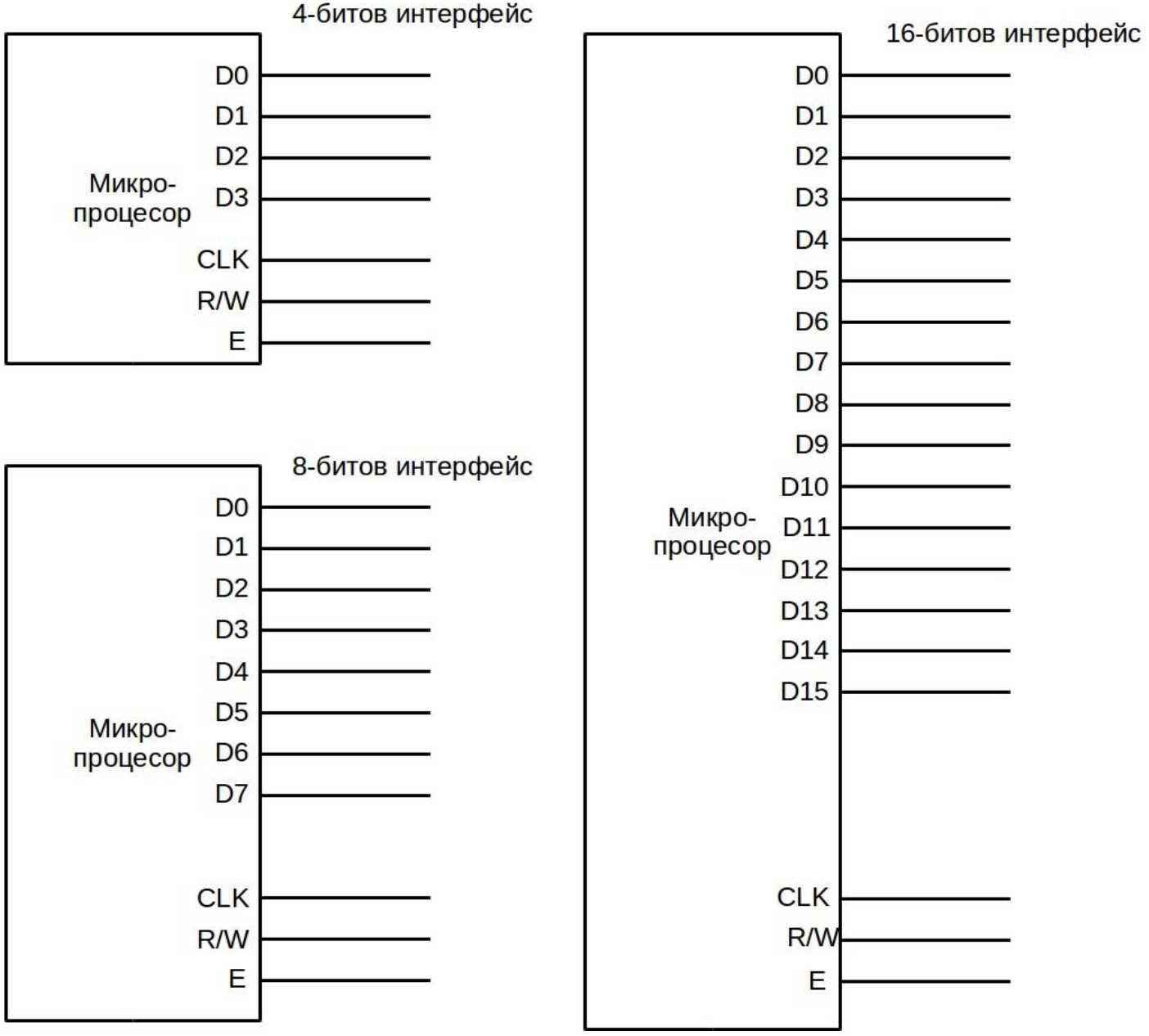
Паралелни асинхронни/синхронни интерфейси

Броят на битовете, които могат да се предадат за един такт, определя **разредността** на интерфейса.

Най-често се използват 4-, 8-, 16-, 32- или 64-битови паралелни интерфейси.

Допълнително към данновите линии има и управляващи, чиито брой зависи от конкретното приложение.

Паралелни асинхронни/синхронни интерфейси



Паралелни асинхронни/синхронни интерфейси

Паралелните интерфейси могат да се разделят на:

***еднопосочни** – информация може да се предава само в една посока – или от главното към подчиненото устройство, или от подчиненото към главното устройство.

***двупосочни** – информацията може да се предава в две посоки – от главното към подчиненото устройство и от подчиненото към главното устройство. Задължително трябва да има **поне един управляващ сигнал**, който да указва посоката на предаване. В противен случай е възможно два изхода да се свържат *накъсо*, което ще доведе до късо съединение, ако логическите им нива са противоположни.

Схемотехника на I/O стъпала

Входни стъпала – основните изисквания към тях са:

*високо входно съпротивление

*малък входен капацитет

*съвместимост на логическите нива, в зависимост от използваната технология на ИС (TTL/CMOS) [1]

*използване на тригери на Шмит

!!!ВНИМАНИЕ!!! Изводите, конфигурирани като входове, не трябва да се оставят плаващи! Поради високото входно съпротивление изводът ще действа като антена за смущения и ще се регистрират лъжливи превключвания на входа. Затова винаги трябва да се свързва “издърпващ” резистор, ако даденият извод има шанс да остане несвързан.

Схемотехника на I/O стъпала

Издърпващите резистори биват два вида:

*към захранване (**pull-up резистор**)

*към маса (**pull-down**)

В зависимост от стойността на резистора:

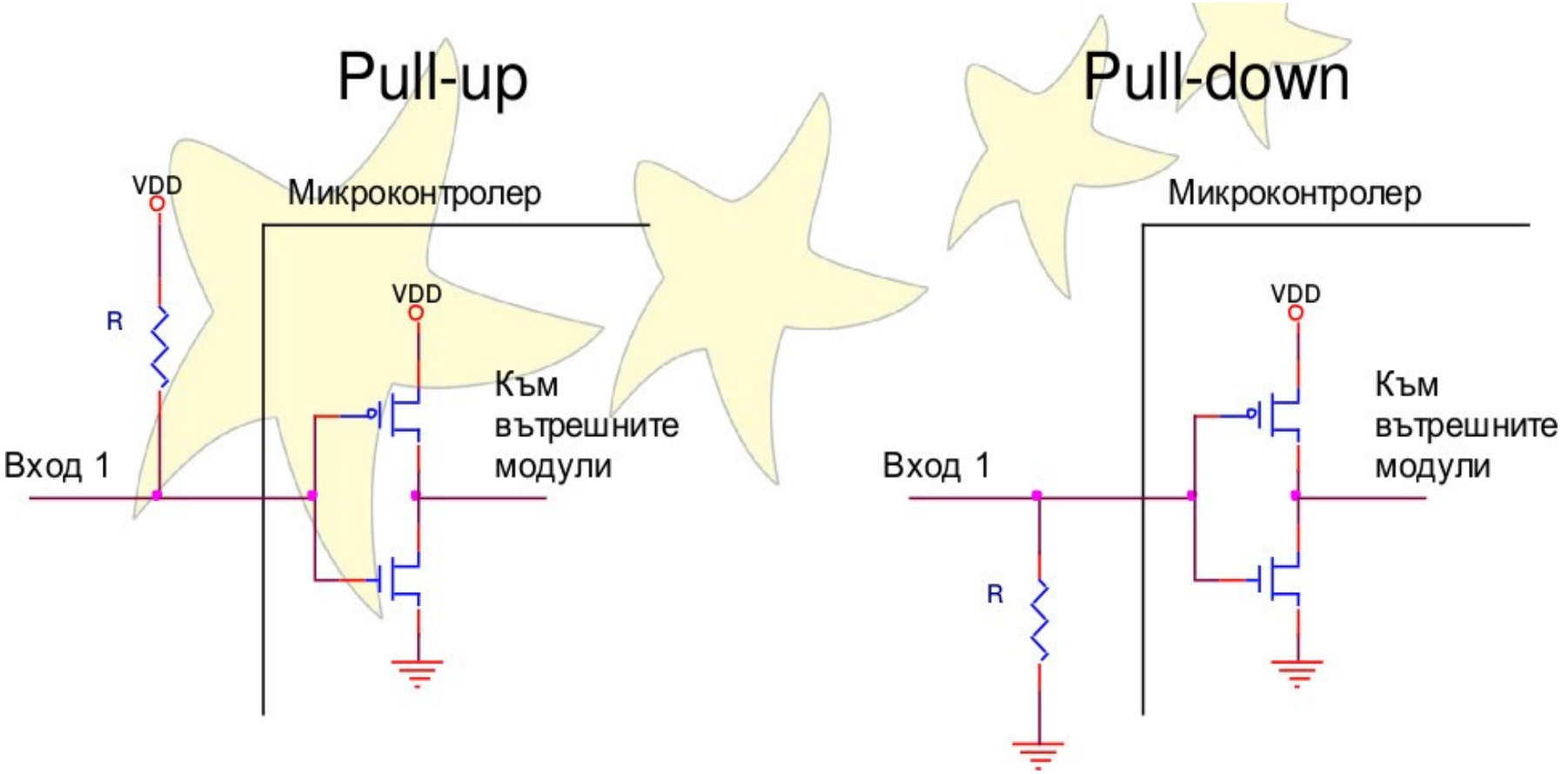
*слаби (**weak** pull-up, weak pull-down) – над 100 kΩ

*силни (**strong** pull-up, strong pull-down) - под 100 kΩ

На самия чип тези резистори се реализират с **генератори на ток**, за да не заемат голяма площ. Токът през генератора определя дали “резистора” ще е слаб или силен.

На схемите на следващият слайд е показано свързването на pull-up и pull-down резистори. Вижда се, че когато изводът е оставен несвързан на входа на инвертора се подава или високо ниво (логическа единица), или ниско ниво (логическа нула) през съответния резистор.

Схемотехника на I/O стъпала

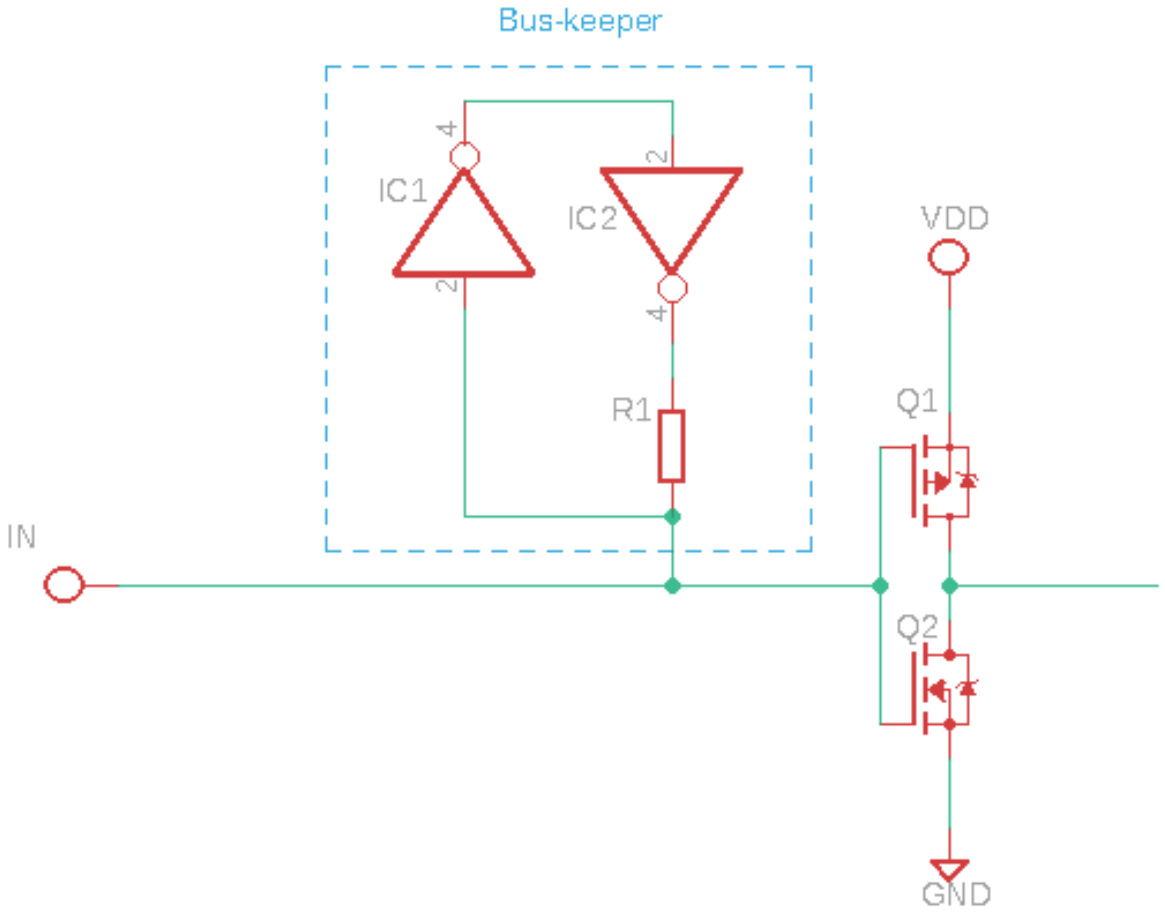


Схемотехника на I/O стъпала

Издърпващите резистори консумират енергия:
*при pull-up това става, когато към входа е подадена логическа нула;
*при pull-down това става, когато към входа е подадена логическа единица.

На следващият слайд е показана една схема, която е подходяща за маломощни приложения. Това е т.нар. **държач на магистрали** (bus-keeper, bus-holder, auto latch circuit), чрез който на входа се задържа последно-установената логическа стойност посредством резистор с двойнствен характер – той е издърпващ и към захранване, и към маса.

Схемотехника на I/O стъпала



Схемотехника на I/O стъпала

!!!ВНИМАНИЕ!!! Високото изходно съпротивление на bus keeper-а може да направи делител с високоомни стъпала/високоомни издърпващи резистори, свързани към входа.

В точката VIN тогава няма да може да се установят **силни логически 1 и 0**.

Схемотехника на I/O стъпала

Пример –

*VDD = 3.3V

*bus keeper out = 0V.

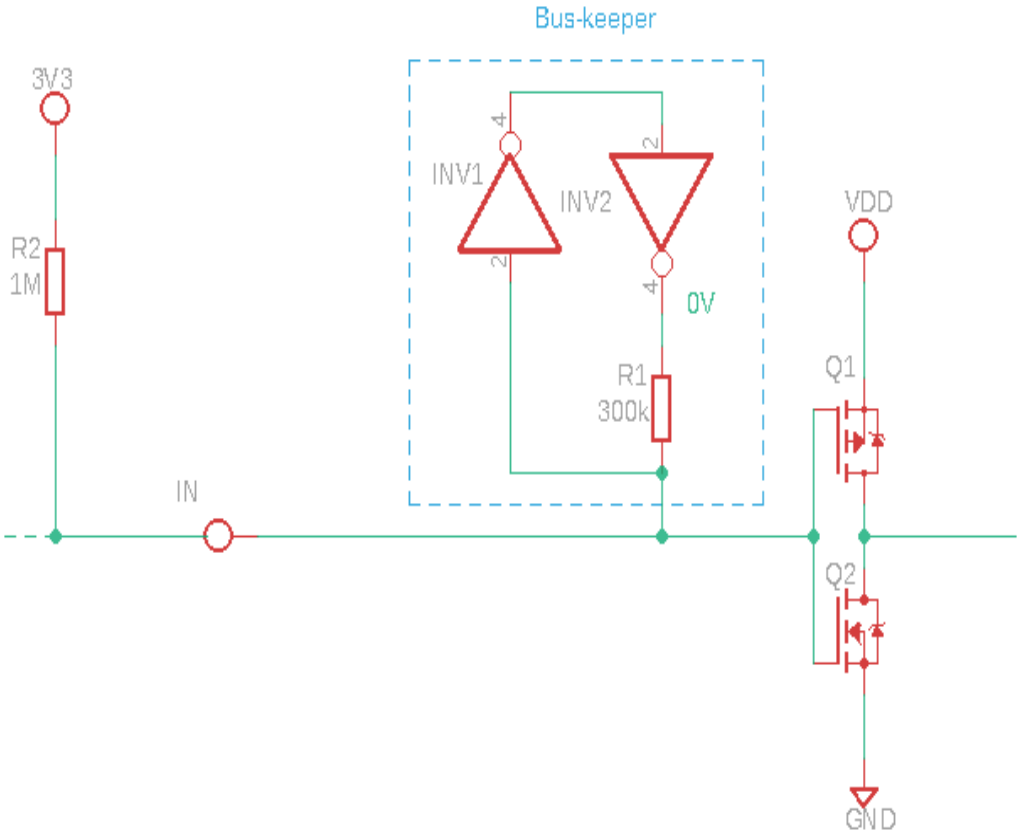
*R1 = 300 kΩ

*R2 = 1 MΩ (pull-up)

=> Iделител = 2.5 μA

=> **V(IN)** = $2.5 \times 10^{-6} \times 300 \times 10^{3}$ **= 0.75V**

=> INV1 няма да превключи.



Схемотехника на I/O стъпала

Съпротивлението на резистора е голямо, за да може при подаване на сигнал от външна схема, мощното ѝ изходно стъпало да вземе превес над държача и да преобърне логическото ниво. Така практически се консумира ток само при преходите от едно ниво в друго (=закъснението на двата инвертора). Когато липсват преходи, схемата консумира статична мощност много по-малка от схемата с издърпващ резистор.

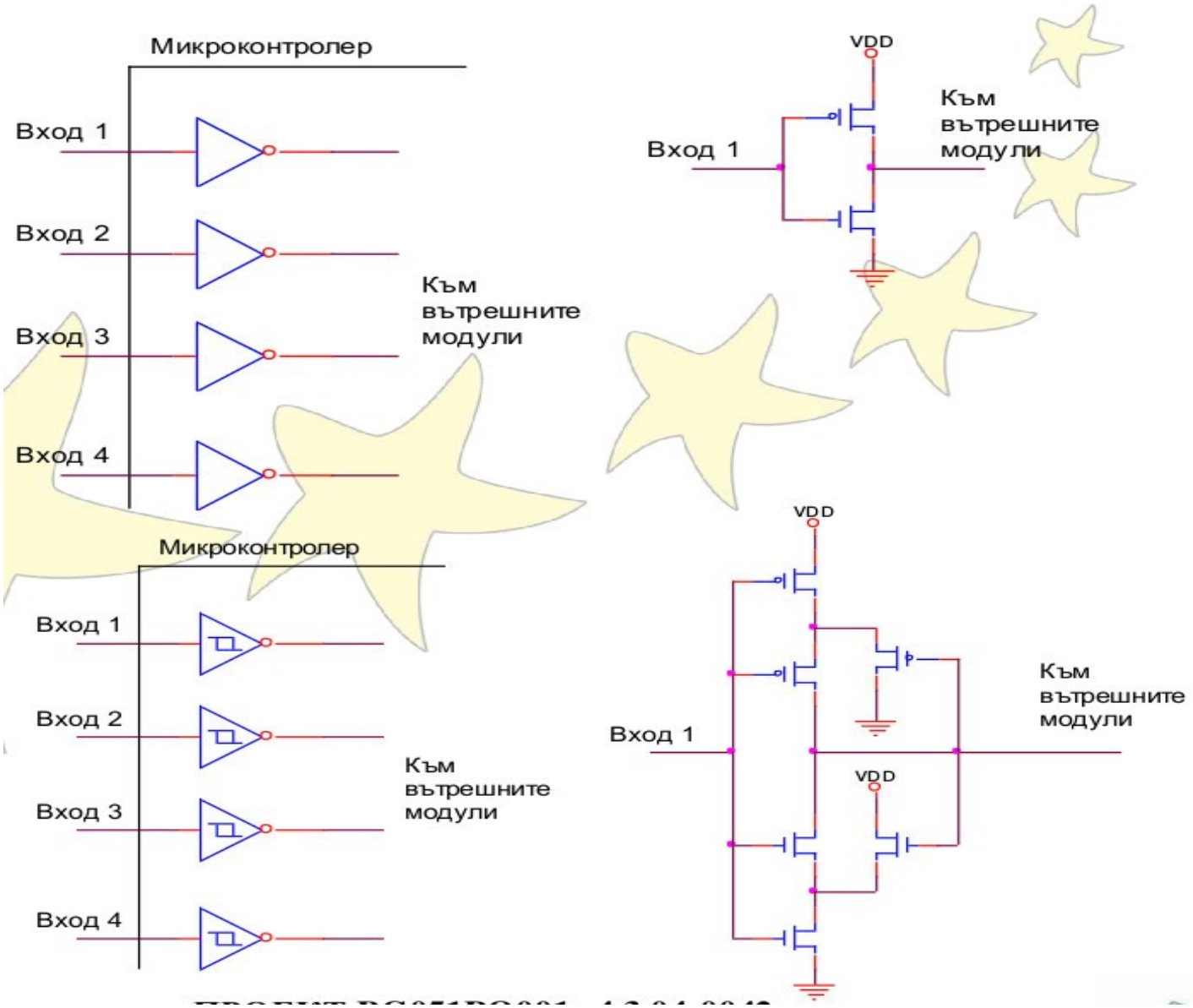
Схемотехника на I/O стъпала

Цифровите входове на микроконтролерите винаги се буферират с инвертиращи или неинвертиращи схеми. Тук говорим за електрическо или **хардуерно буфериране** на вход, което позволява постигането на високо входно съпротивление, малък входен капацитет и въвеждане на хистерезис.

Съществува и буфериране на входен порт чрез паралелен регистър, което засяга четенето на данните от него. Това се нарича **софтуерно буфериране**.

Хардуерно буфериране чрез инвертори с и без тригер на Шмит е показано на схемите в следващия слайд. Дадена е принципната схема на един от инверторите.

Схемотехника на I/O стъпала

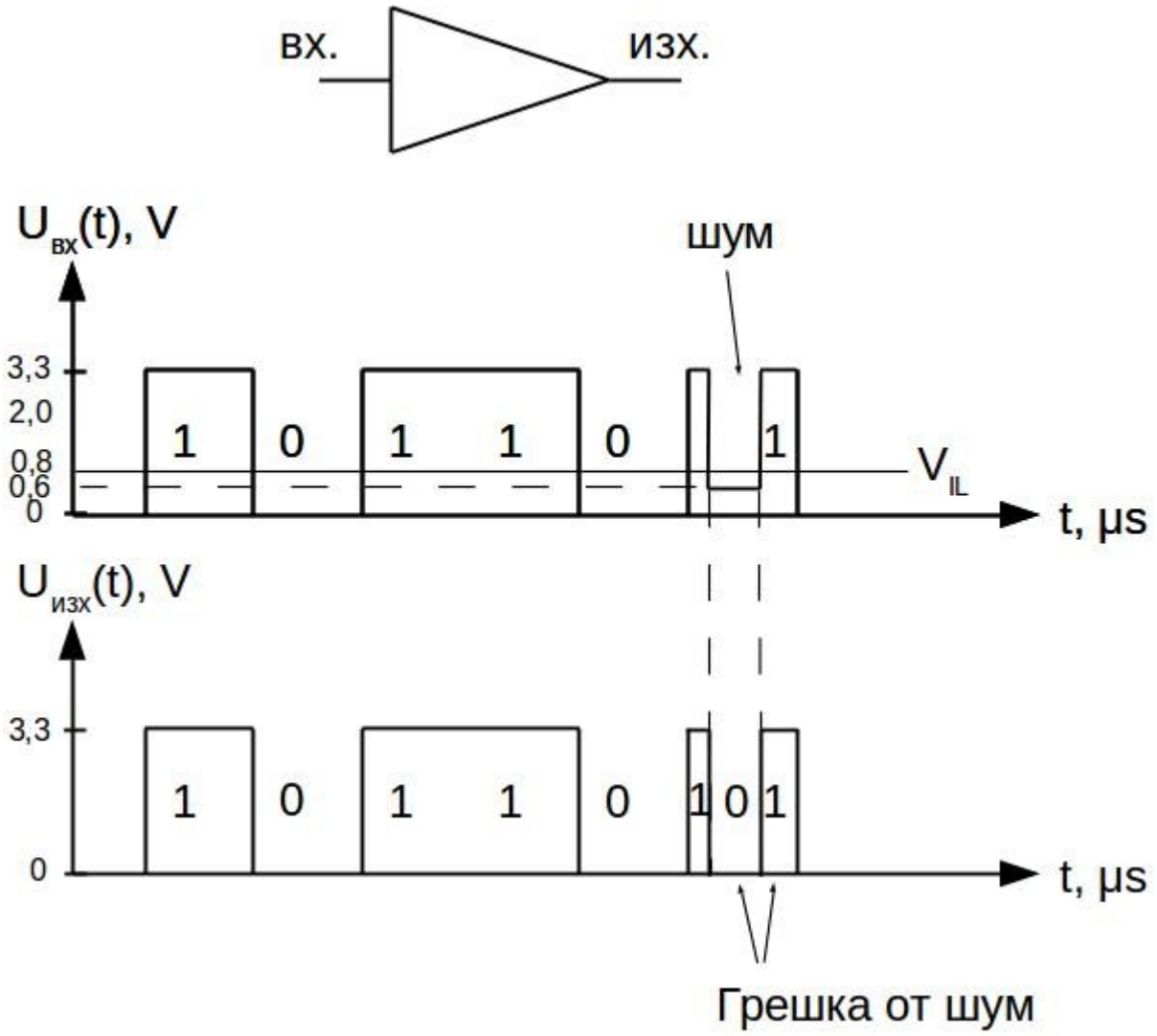


Схемотехника на I/O стъпала

Използването на **тригери на Шмит** при входовете подобрява **шумоустойчивостта** им.

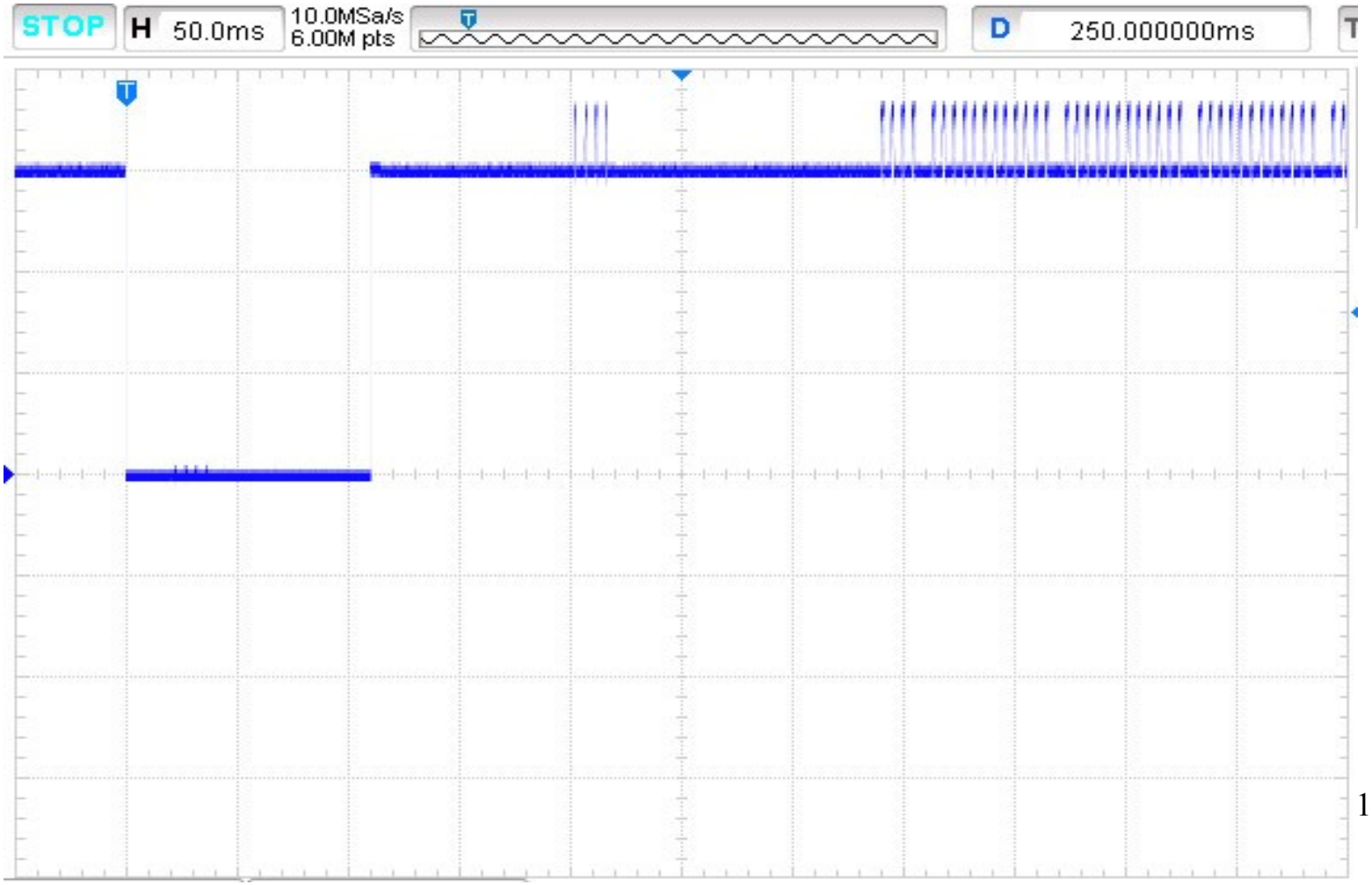
На фигурите на следващият слайд е даден пример за използване на вход без и с тригер на Шмит. Входният сигнал е изкривен поради смущения, което води до лъжливо превключване на входния буфер без тригер, а оттам – грешно приемане на данни от микропроцесора.

Схемотехника на I/O стъпала



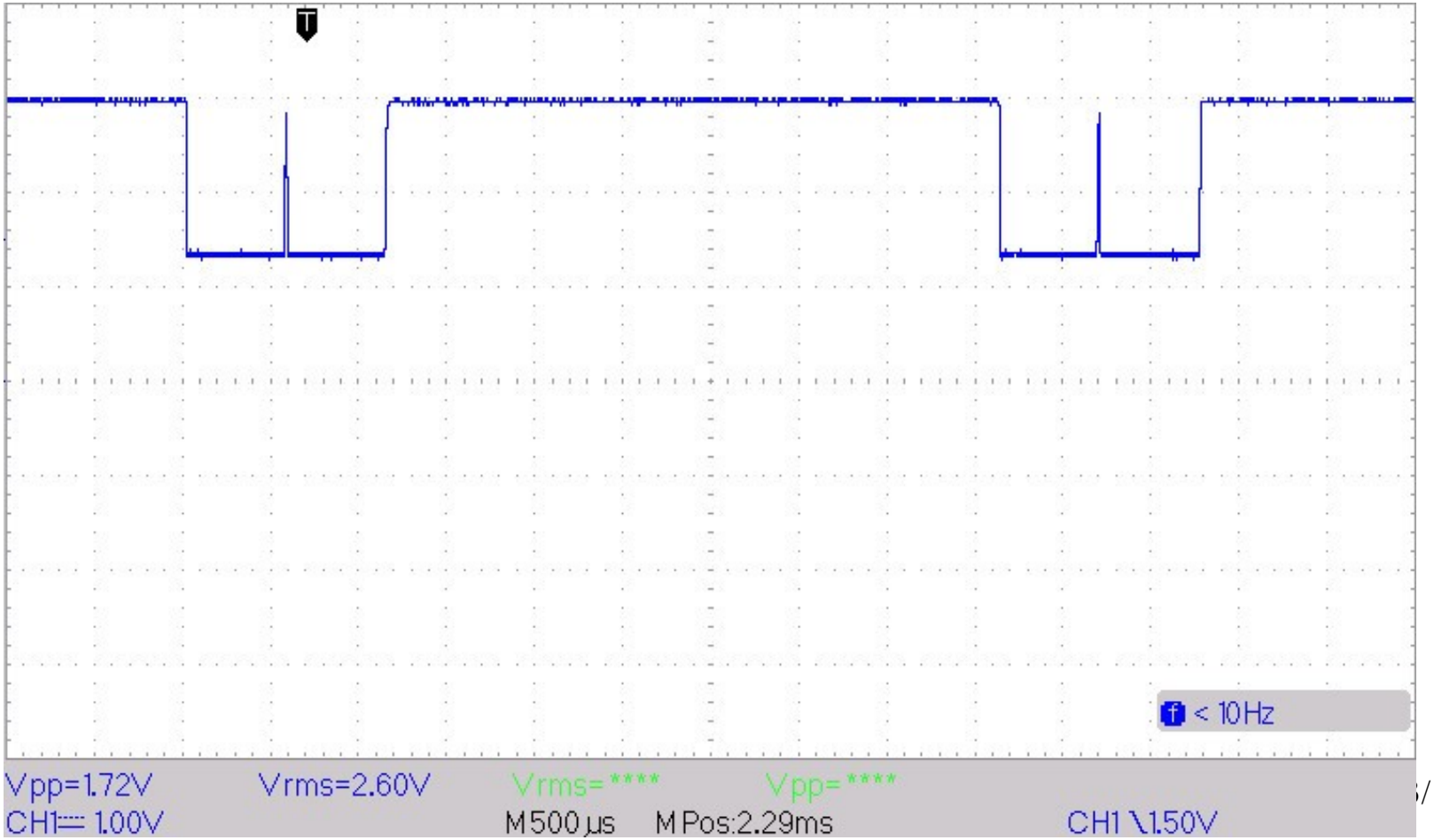
Схемотехника на I/O стъпала

Пример – на осцилограмата по-долу е даден сигналът на вход на микропроцесор в близост до GSM модул. Вижда се насложеното смущение, което в случая е с положителен знак.



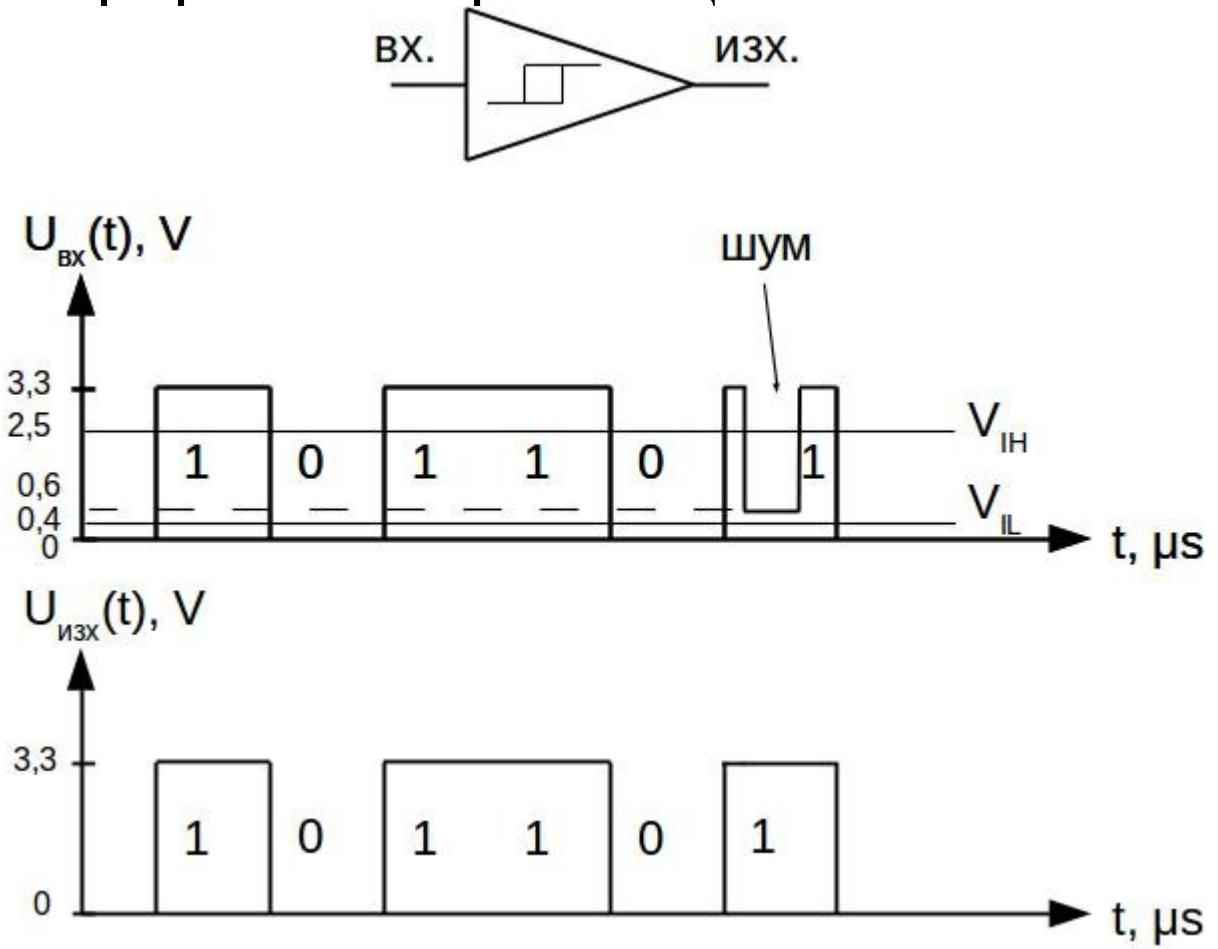
Схемотехника на I/O стъпала

Пример – на осцилограмата по-долу е даден сигналът на вход на микропроцесор в близост до GSM модул. Вижда се насложеното смущение, което в случая е с отрицателен знак.



Схемотехника на I/O стъпала

Тригерите на Шмит позволяват да се въведе долен и горен праг на възприемане на логическите нива. Сигнали по-големи от долния праг и по-малки от горния праг няма да бъдат регистрирани от приемащата част.



Схемотехника на I/O стъпала

Някои микроконтролери имат схеми за **филтриране на смущения** (debounce, deglitch suppresion) на входовете чрез използване на хардуерно времезакъснение. Времезакъснението се реализира с програмируем брояч.

Пример – микроконтролерът Sitara AM3353 (TI) има два регистъра за управление на тази функционалност:

GPIO_DEBOUNCEENABLE – разрешаване на хардуерното времезакъснение за съответен извод (има по един бит за всеки извод): 1 – разрешено, 0 - забранено
GPIO_DEBOUNCINGTIME – първите 8 бита определят коефициент N, който участва във формулата:

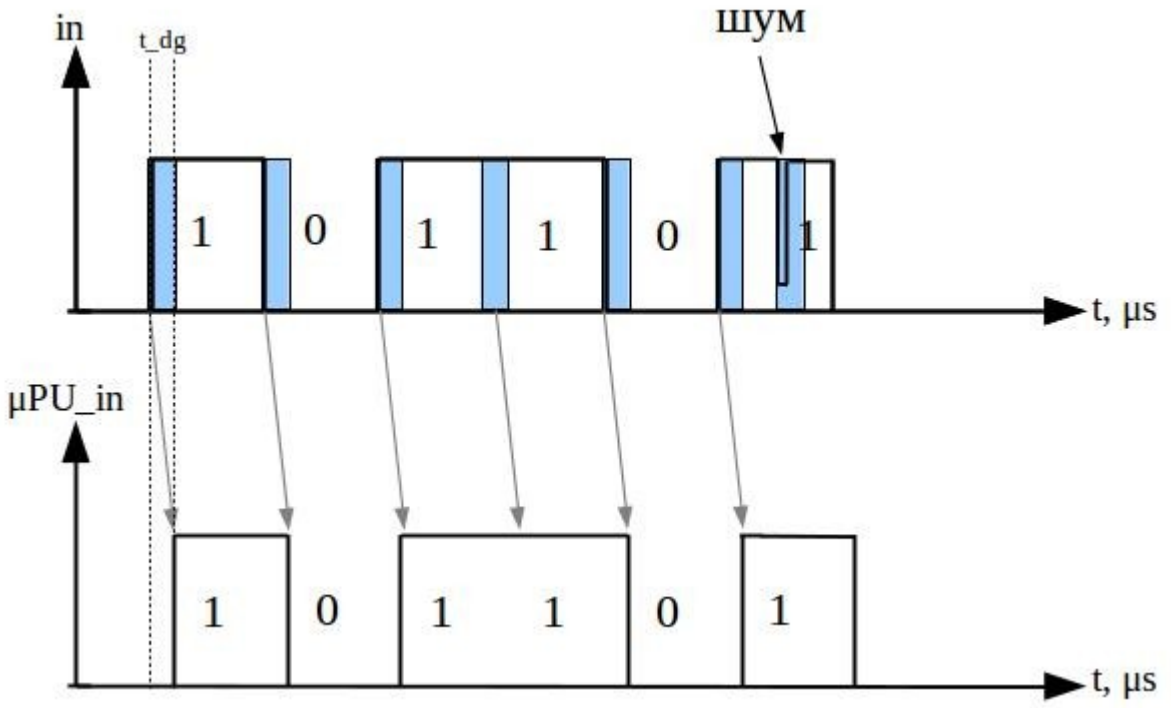
t_delay = (N+1)*31 = [us]

Схемотехника на I/O стъпала

Пример – микроконтролерът MSP430G2553 (TI) има един регистър за управление на хардуерно времезакъснение:

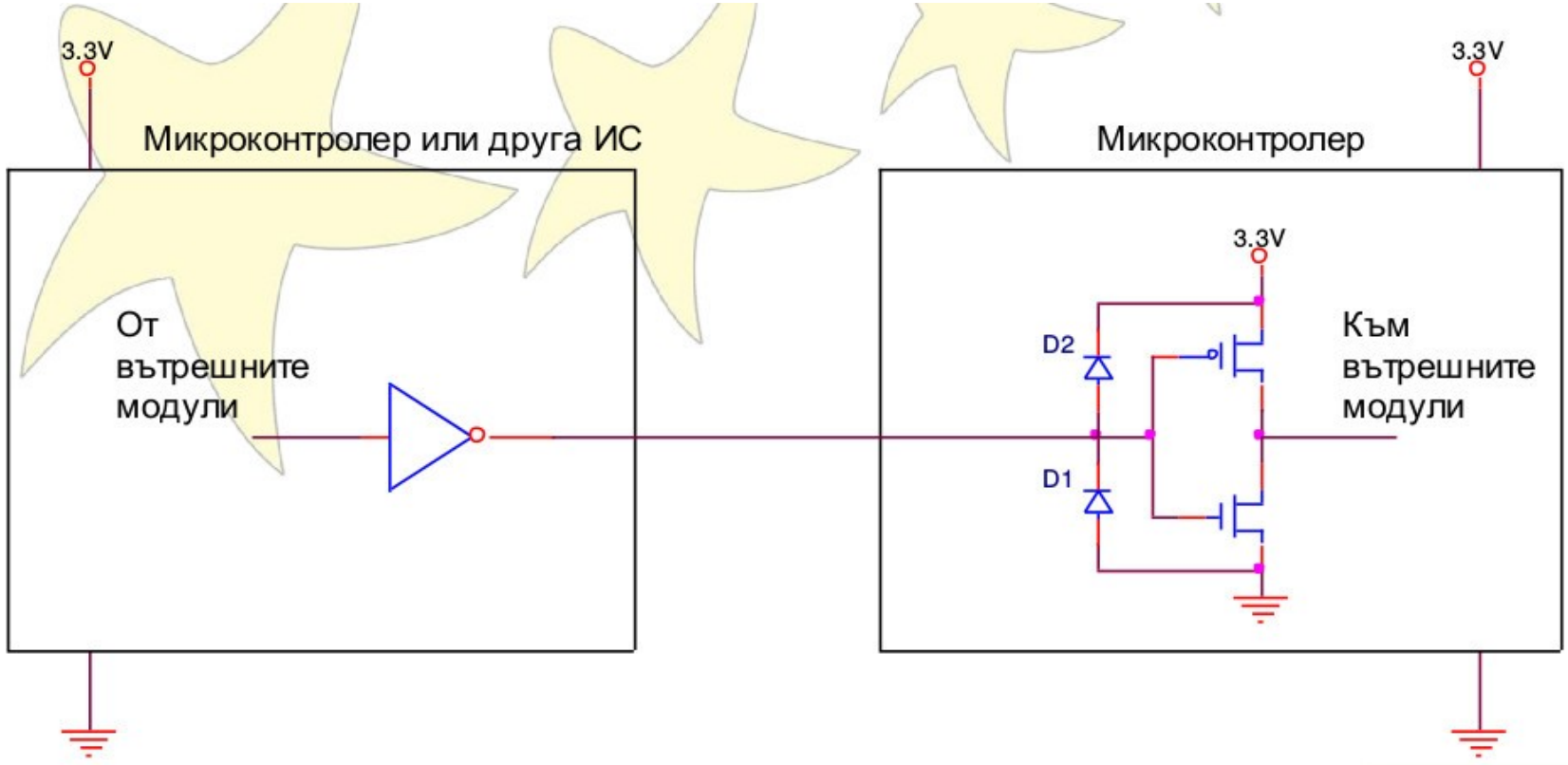
UCBxCTLW1 – първите два бита от този регистър (наречени UCGLIT0 и UCGLIT1) избират предефинирани стойности на времето:

00 – 50 ns
01 – 25 ns
10 – 12.5 ns
11 – 6.25 ns



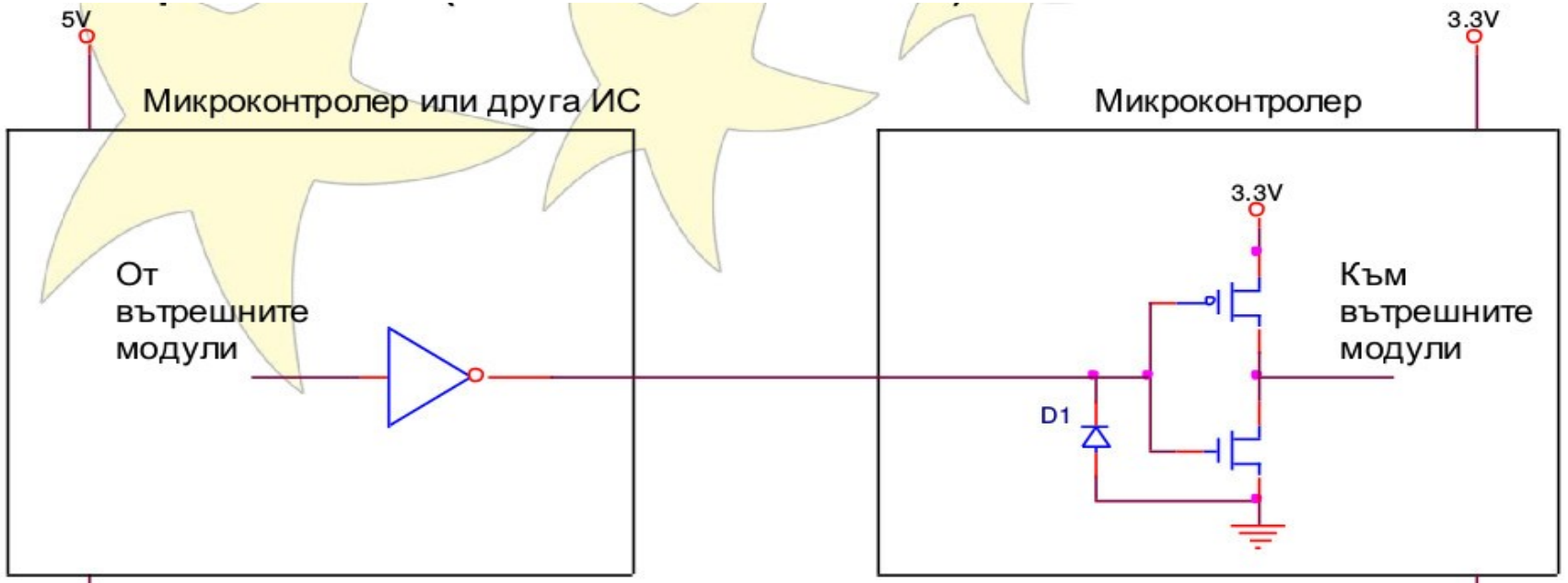
Схемотехника на I/O стъпала

Входните стъпала на микроконтролерите обикновено имат **защити по напрежение**. В случая диод D1 предпазва входа от отрицателни напрежения, а D2 от положителни напрежения, по-високи от захранващото.



Схемотехника на I/O стъпала

5V-толерантни входове – това са входове на микроконтролер, който се захранва с напрежения по-ниски от 5 V (например с 3.3 V), но схемотехниката позволява към изводите му да се свързват интерфейси от микроконтролер (или друга ИС), работещ на 5 V без да настъпи повреда. При тях се слага само D1. входните транзистори се проектират така, че да понесат по-високи напрежения (до около 6 – 7 V).



Схемотехника на I/O стъпала

Пример – микроконтролерът TI Sitara AM3353 има два регистъра за управление на тази функционалност:

GPIO_DEBOUNCEENABLE – разрешаване на хардуерното времезакъснение за съответен извод (има по един бит за всеки извод): 1 – разрешено, 0 - забранено
GPIO_DEBOUNCINGTIME – първите 8 бита определят коефициент N, който участва във формулата:

t_delay = (N+1)*31 = [us]

Схемотехника на I/O стъпала

Изходни стъпала – основните изисквания към тях са:

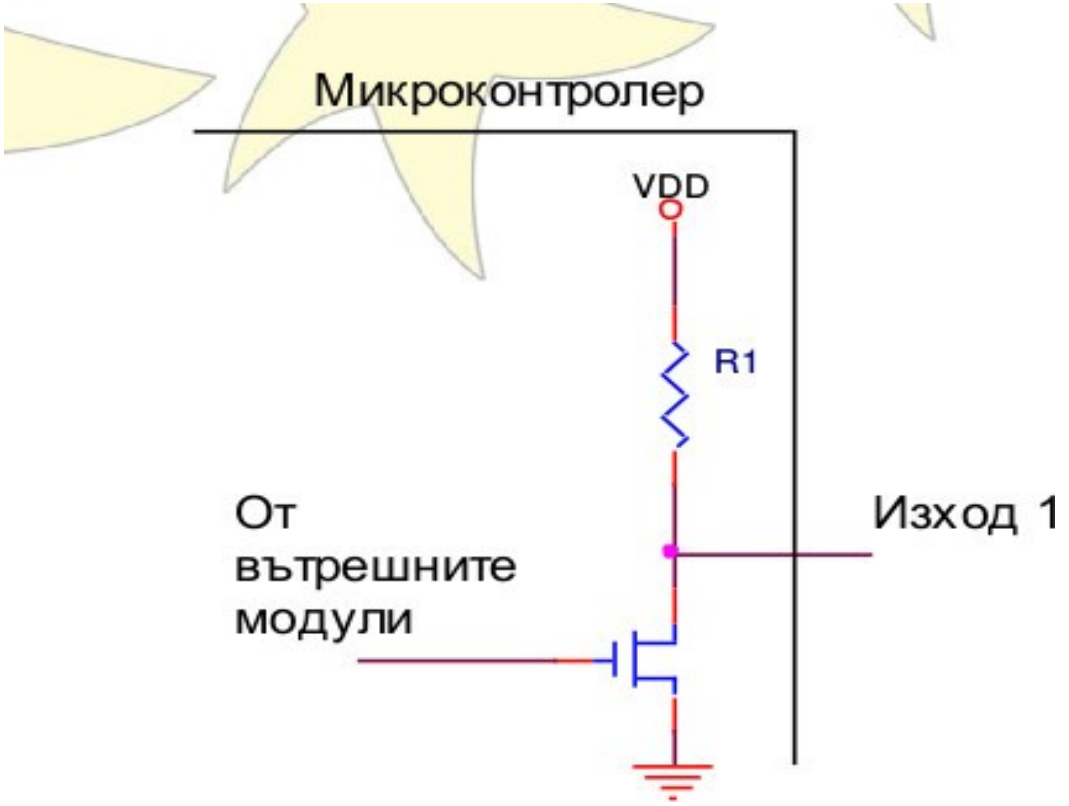
*голям изходен ток, I outmax (ниско изходно съпротивление)

*висока честота на превключване

*съвместимост на логическите нива, в зависимост от използваната технология на ИС (TTL/CMOS) [1]

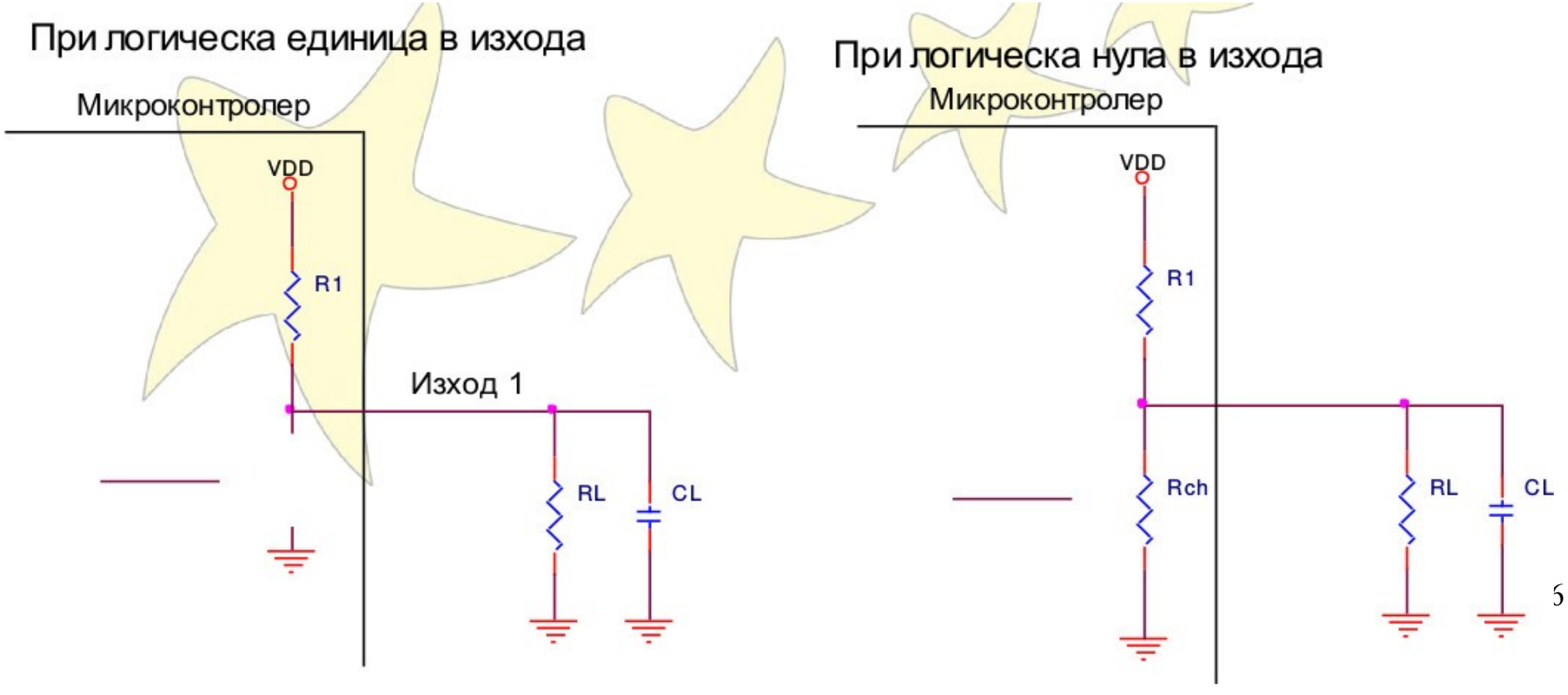
Схемотехника на I/O стъпала

Изходни стъпала с режимен резистор – в по-старите интегрални схеми (PMOS и след това NMOS логика) транзисторните ключове са били реализирани по схемата, показана по-долу:



Схемотехника на I/O стъпала

При тази схема на свързване изходният ток ще се определя от съпротивлението на резистора R1 при логическа единица в изхода и от съпротивлението на канала на NMOS транзистора при логическа нула. Показани са еквивалентните схеми на стъпалото в двата случая.

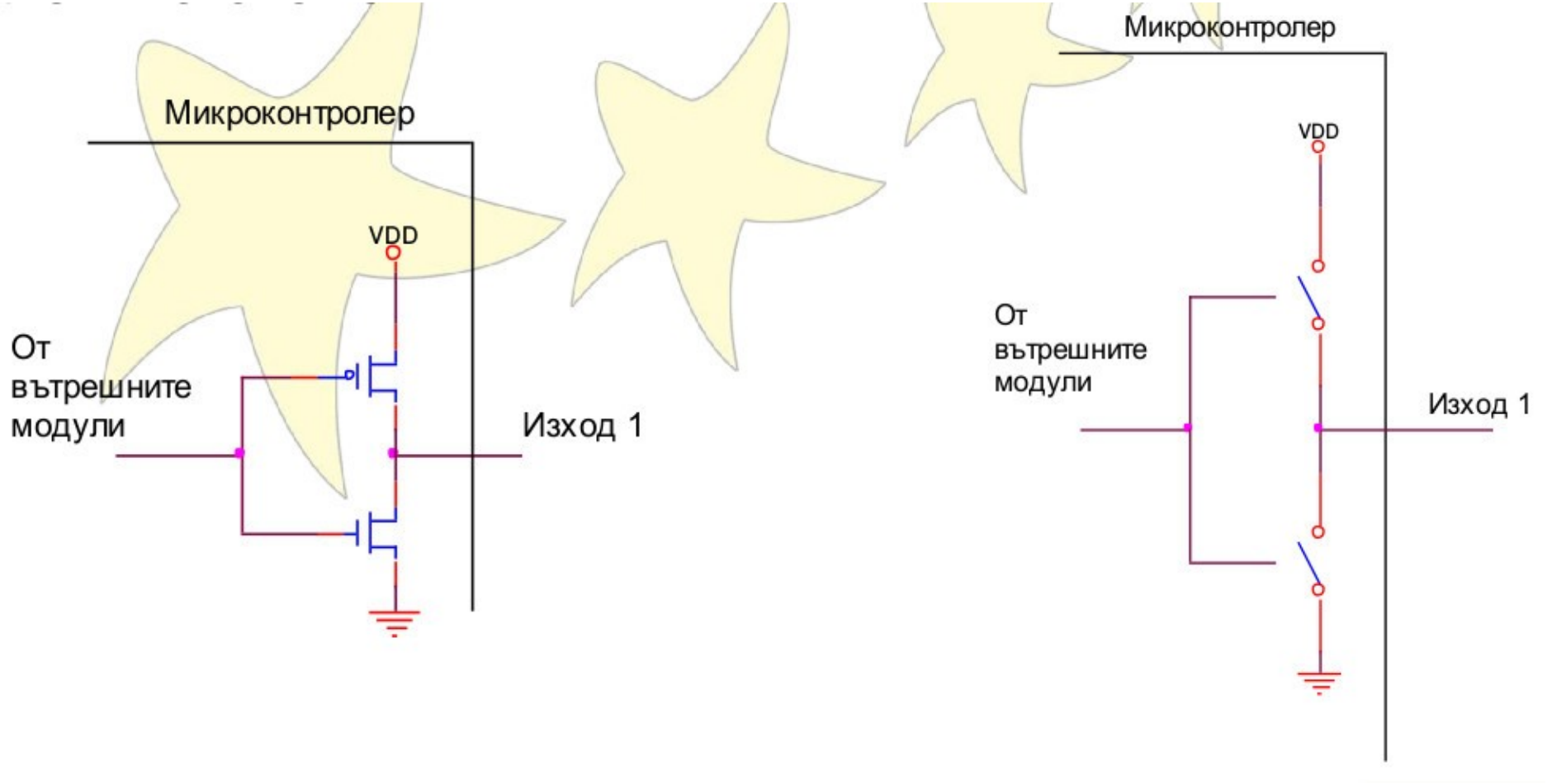


Схемотехника на I/O стъпала

Противотактни (push-pull) изходни стъпала – използват се в съвременните интегрални схеми. Този вид логика се нарича комплементарна (допълваща се) MOS логика или още - CMOS. При тях режимният резистор е заменен с друг транзистор и така при установяване в логическа единица статична консумация на мощност няма (ако пренебрегнем утечните токове).

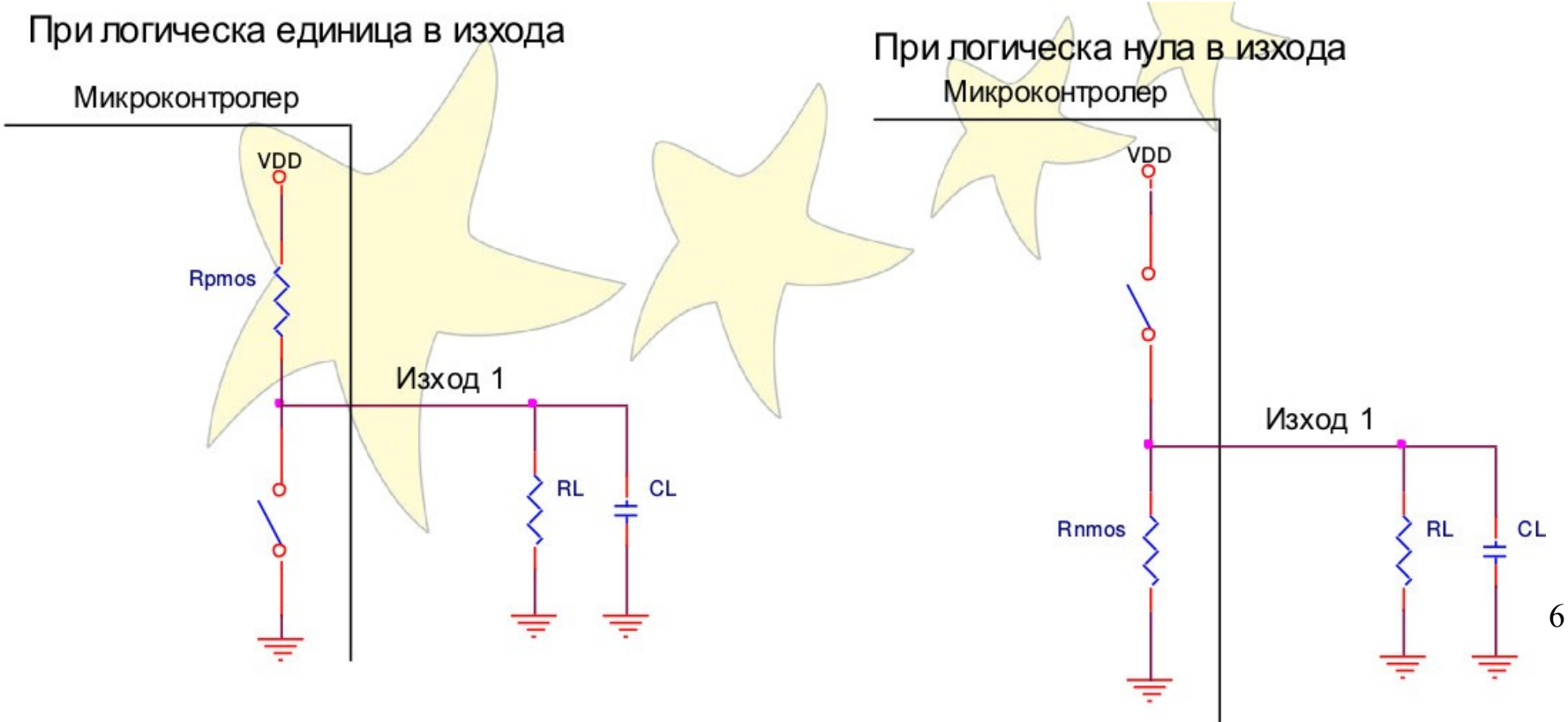
На фигурите на следващия слайд е даден инвертиращ изходен буфер и неговата еквивалентна схема.

Схемотехника на I/O стъпала



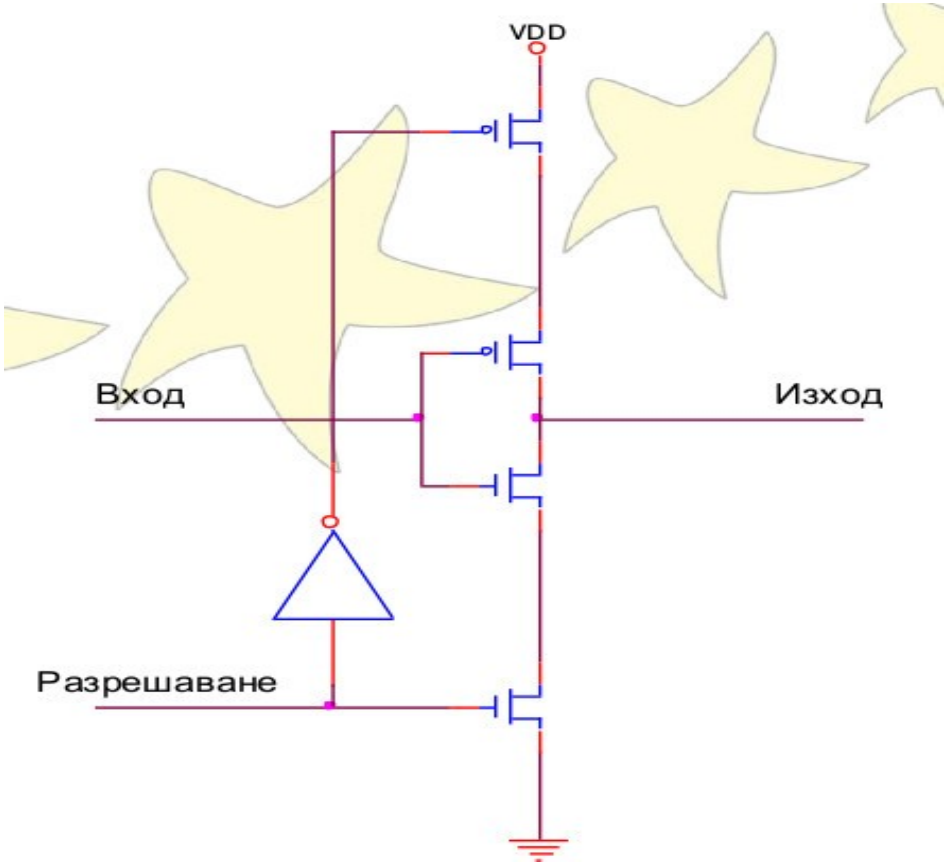
Схемотехника на I/O стъпала

Товароносимостта на изходното стъпало зависи от съпротивлението на канала на P- и NMOS транзисторите. Понеже е възможно R_DS_nmos да е различно от R_DS_pmos , то максималния ток, който ще може да се осигури от изхода, ще е различен за логическата нула и единица.



Схемотехника на I/O стъпала

Противотактните стъпала позволяват реализирането на буфер с **високоимпедансно състояние**, при което изхода не е свързан нито към логическа единица, нито към логическа нула. Такива буфери са много полезни при двупосочна комуникация между повече от две устройства.



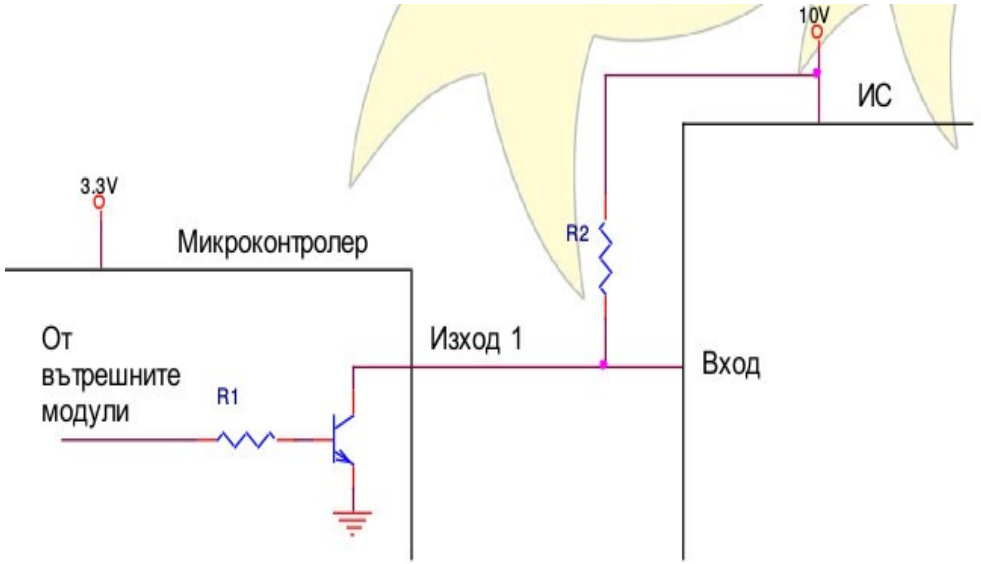
Схемотехника на I/O стъпала

Изходни стъпала с отворен колектор/дрейн – при тях липсва както режимно съпротивление, така и PMOS транзистор.

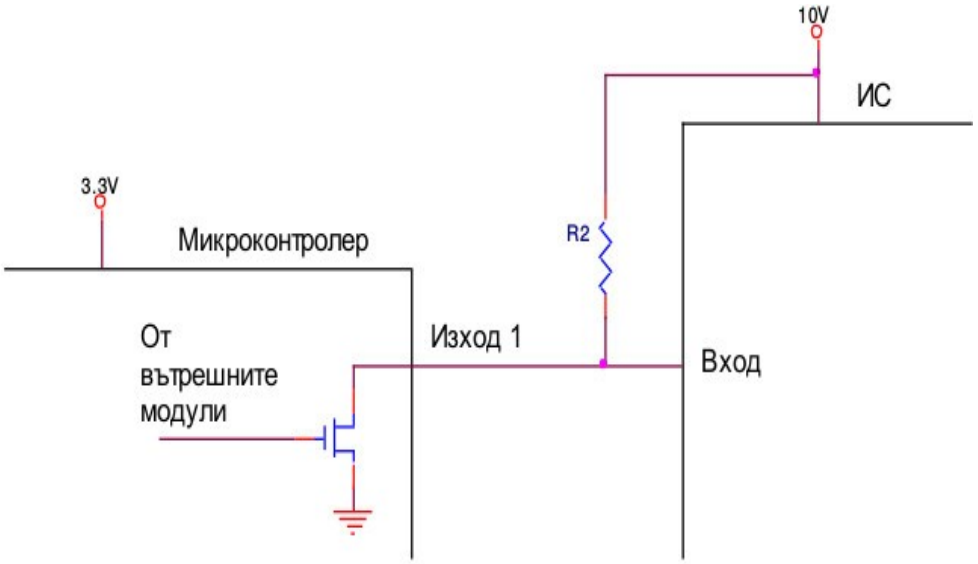
Свързването на външно режимно съпротивление или друг товар е оставено на проектанта. Това позволява **транслиране** на логическите нива между различни серии.

На следващият слайд е показано предаване на информация между ИС със захранващо напрежение 3.3 V и 10 V. При такова свързване има ограничение – захранващото напрежение, към което ще бъде свързан отворения колектор/дрейн не трябва да надхвърля напрежението U_CEmax или U_DSmax на съответния транзистор.

Схемотехника на I/O стъпала



Изход от тип отворен колектор



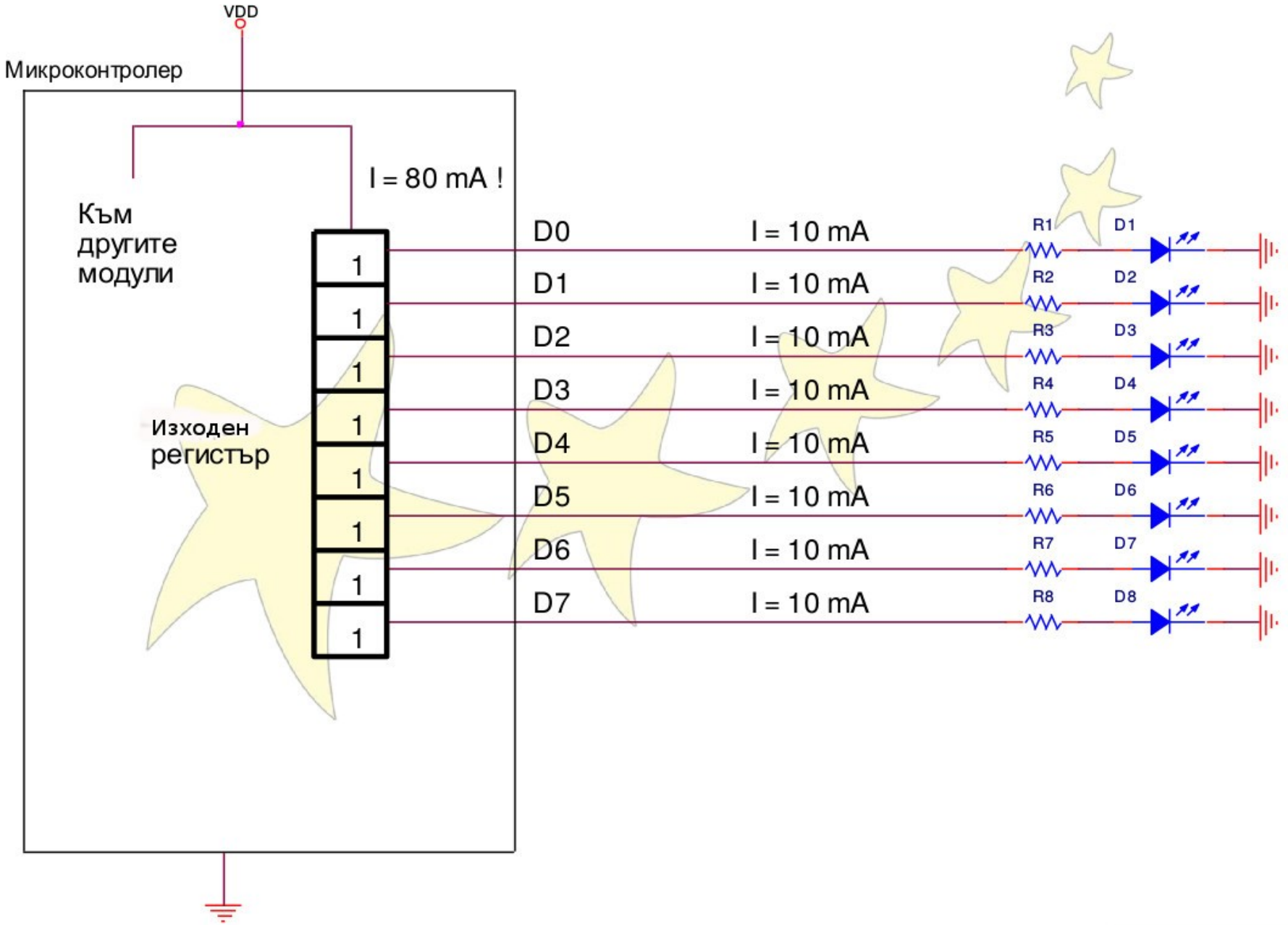
Изход от тип отворен дрейн

Схемотехника на I/O стъпала

При свързването на товари към изходните стъпала **трябва да се съобразява максималния ток**, който може да се осигури от даден извод без да настъпи повреда в ИС. Освен това при проектирането трябва да се съобрази и максималния ток, който може да протече през порта, в който се намира даденият извод.

Пример - на фигурата в следващия слайд се използва микроконтролер, който може да осигури на всеки свой извод до 10 mA ток. Максималният ток, който може да осигури един порт от 8 извода е 60 mA и тогава показаното свързване ще доведе до повреда на микроконтролера при едновременното включване на повече от 6 светодиода!

Схемотехника на I/O стъпала

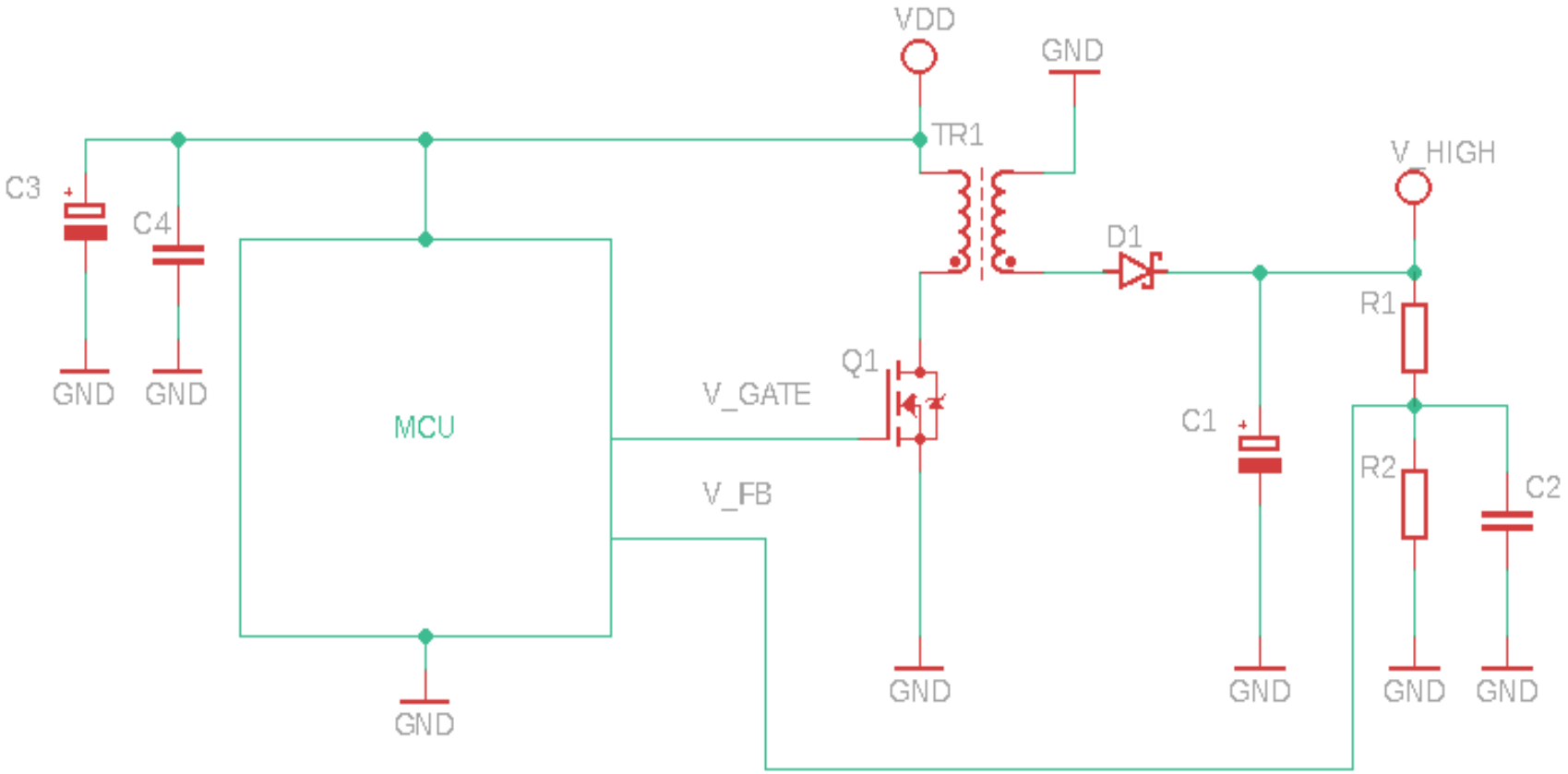


Схемотехника на I/O стъпала

При свързването на товари към изходните стъпала **трябва да се съобразява работната честота.** Ако е свързан MOS транзисторен ключ, капацитетът C_gs ще окаже влияние върху фронтовете на превключване, а оттам и върху енергийната ефективност на схемата.

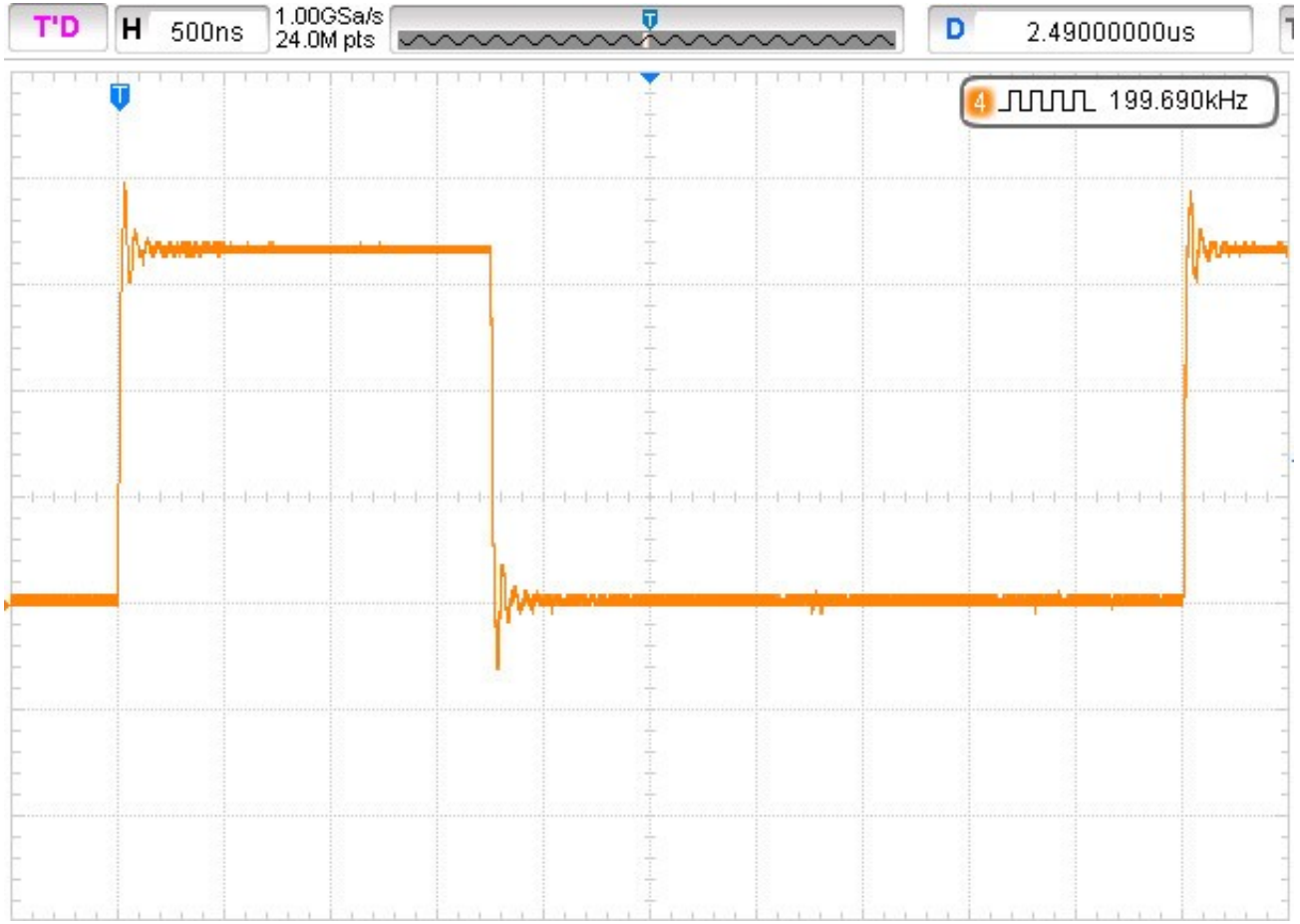
Пример - на схемата от следващия слайд се използва микроконтролер, който участва в контролната верига на повишаващ постояннотоков преобразувател. Товароносимостта на изходното стъпало може да бъде конфигурирано на 6 и 25 mA (за конкретния MCU). Със стъпало 25 mA КПД-то на схемата се подобрява с 30 – 50 % спрямо варианта с 6 mA. В случая е важно **софтуерът** да конфигурира правилно съответния извод.

Схемотехника на I/O стъпала



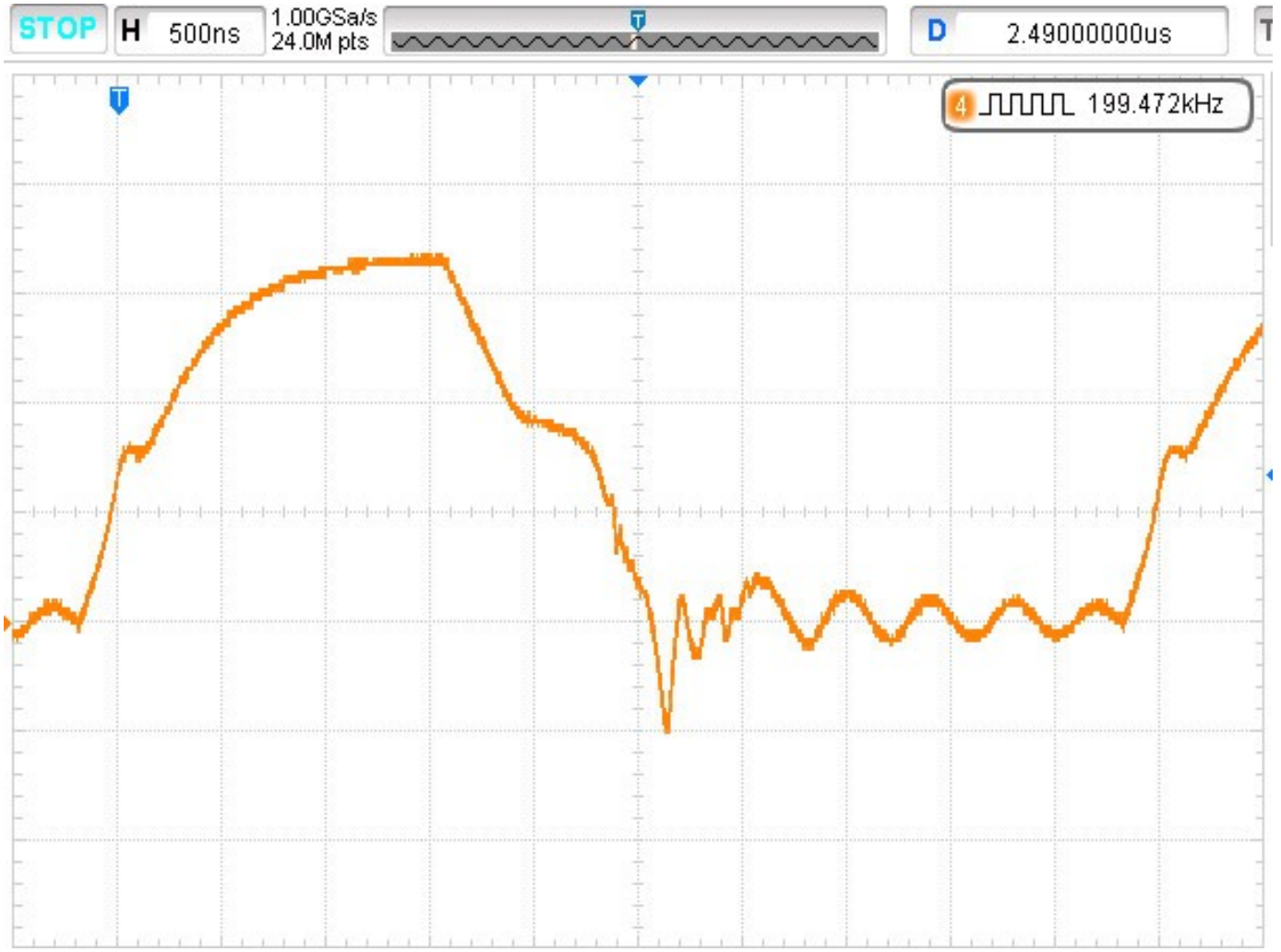
Схемотехника на I/O стъпала

Осцилограма на сигналът V_GATE без да е свързан NMOS транзисторът Q1.



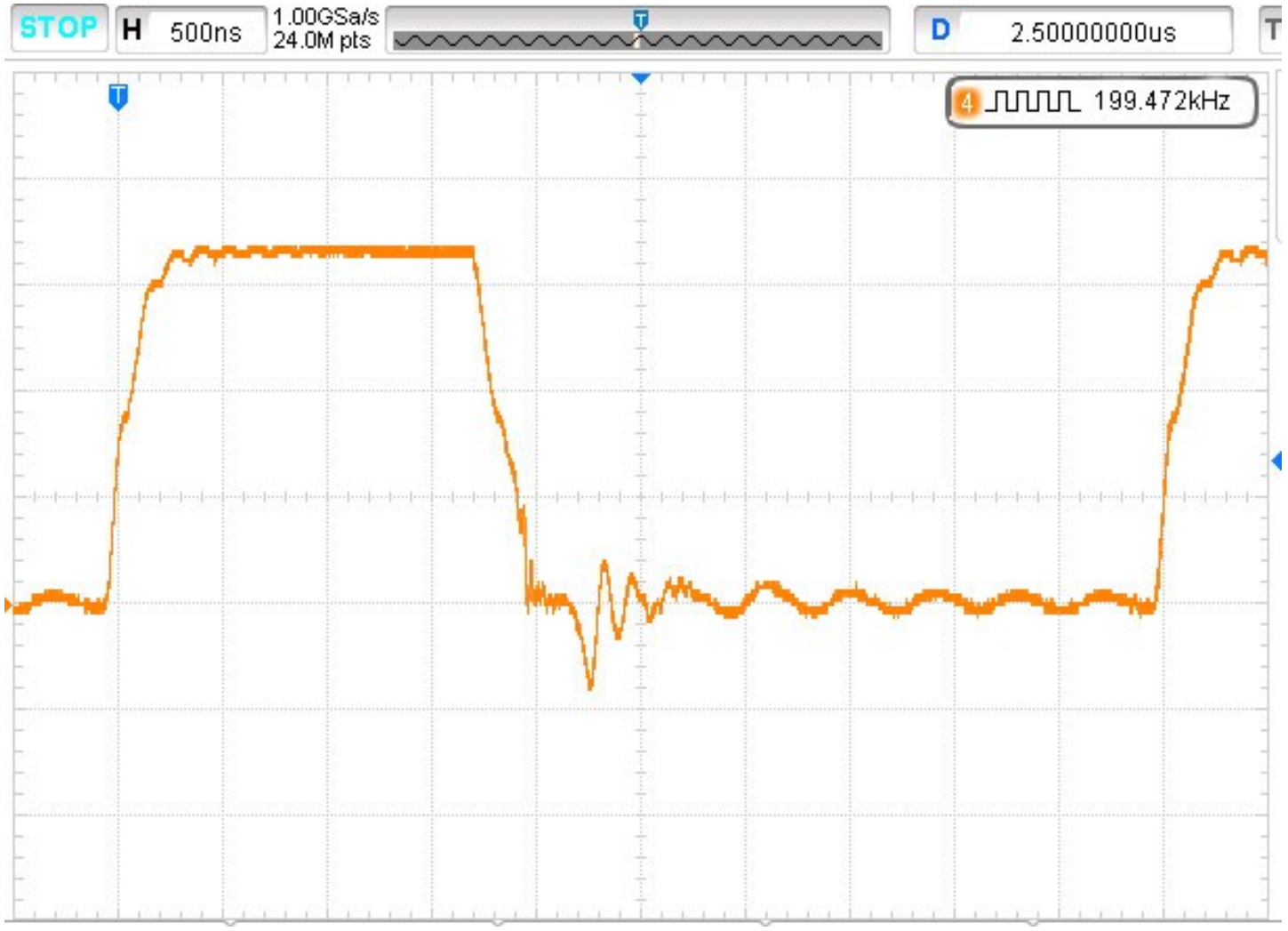
Схемотехника на I/O стъпала

Осцилограма на сигналът V_GATE, със свързан Q1, с товароносимост на изходното стъпало 6 mA.



Схемотехника на I/O стъпала

Осцилограма на сигналът V_GATE, със свързан Q1, с товароносимост на изходното стъпало 25 mA.



Схемотехника на I/O стъпала

За да може две или повече интегрални схеми с различни захранващи напрежения да работят заедно в една система, трябва връзките между тях да включват **транслатори на нива** (level shifters).

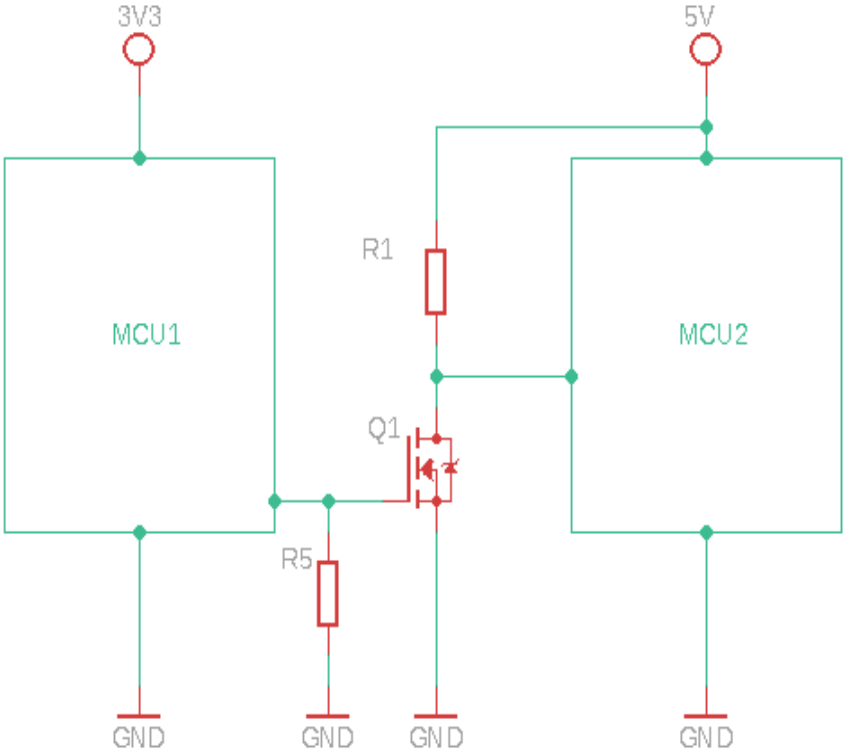
Транслаторите на нива преобразуват амплитудите на сигналите на източника в амплитуди, които приемникът може да понесе.

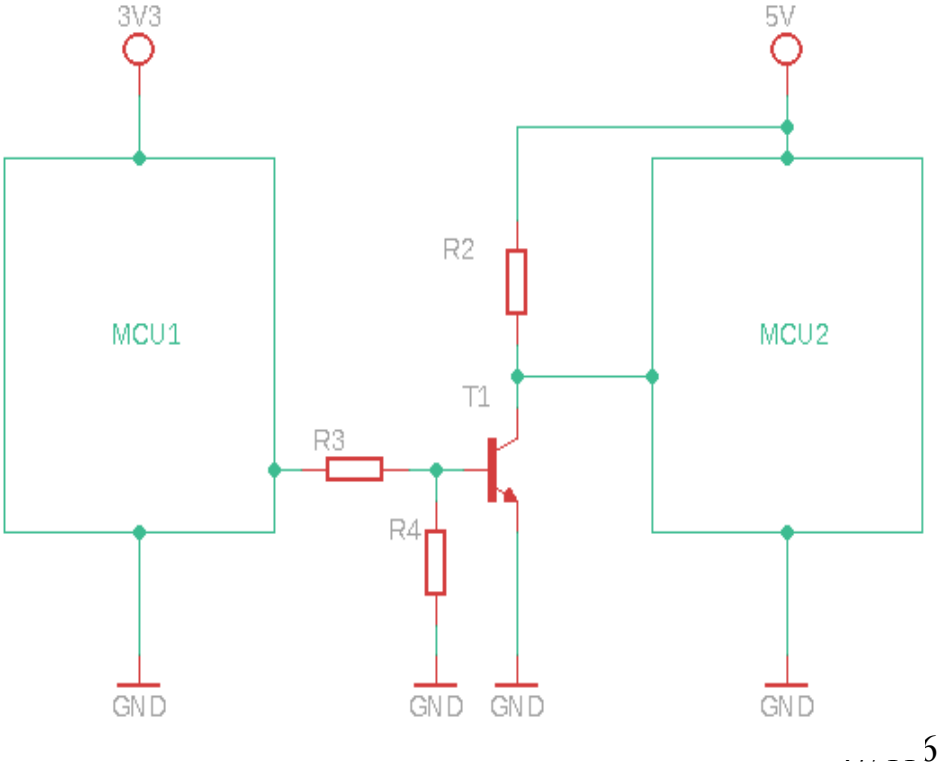
Според схемната реализация съществуват следните видове:

*инвертиращи или неинвертиращи
*галванично разделени или свързани
*еднопосочни или двупосочни

Схемотехника на I/O стъпала

Най-простите и най-неефективните схеми са показани по-долу. R4 и R5 са pull-down резистори, за да не остават гейта и базата плаващи при началното стартиране на системата. R3 задава базовия ток на T1. R1 и R2 са режимни съпротивления.





Схемотехника на I/O стъпала

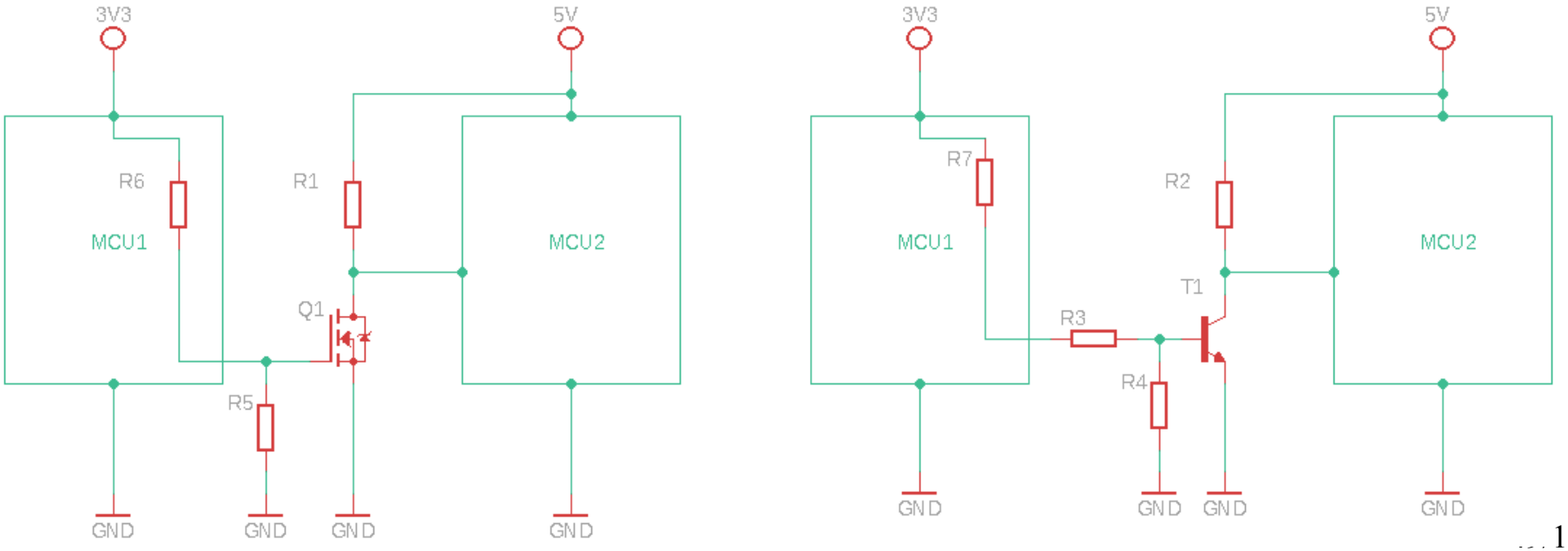
По-ниски стойности на R1 и R2 ще направят нарастващите фронтове по-стръмни, но ще повишат консумацията на схемата.

Фронтовете ще са несиметрични – падащият фронт ще е стръмен (определя се от Rds и r0), а нарастващият полегат, заради високоомните режимни съпротивления.

!!!ВНИМАНИЕ!!! Някои микроконтролери стартират със всички изводи конфигурирани като входове с вътрешни издърпващи резистори. Това може да доведе до фалшиви превключвания при стартиране на системата (виж следващия слайд).

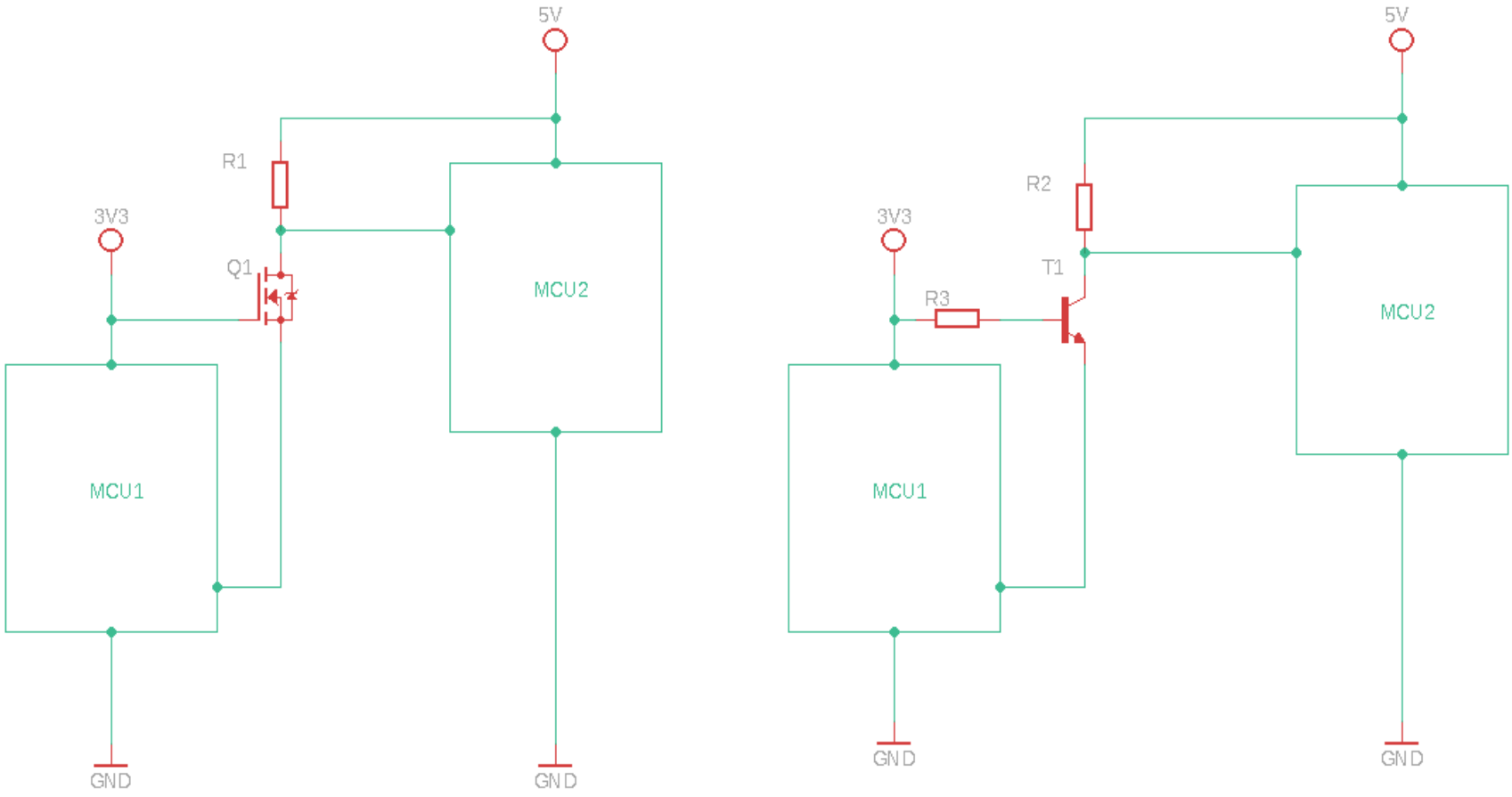
Схемотехника на I/O стъпала

Пример – микроконтролерът LPC824 стартира с всички изводи като входове със свързани издърпващи резистори към захранване. В този случай R6 и R5 ще направят делител, който може да отпуши Q1. При биполярния транзистор – базов ток ще протече през R7 и R3.



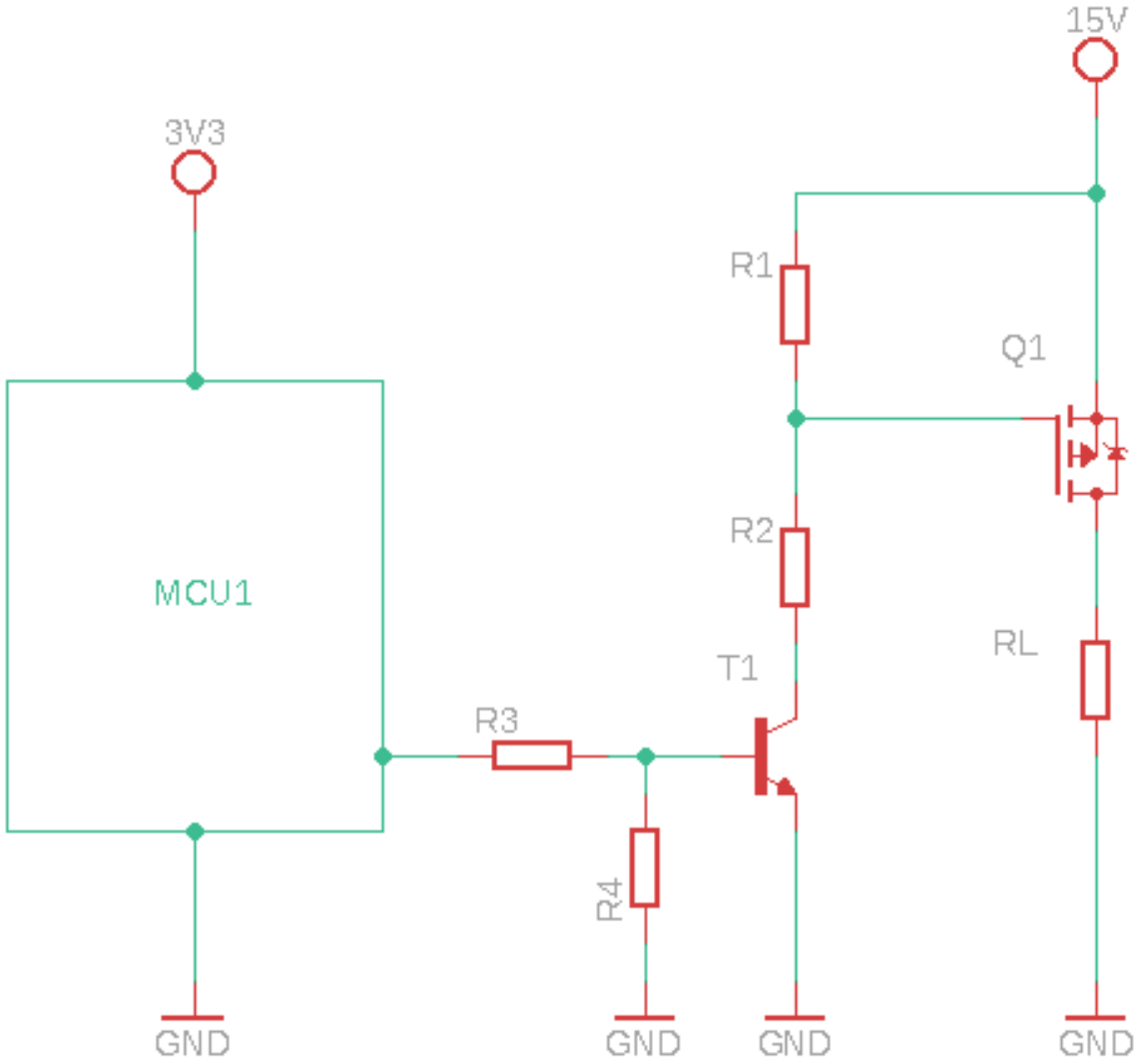
Схемотехника на I/O стъпала

Неинвертиращ вариант на предишните схеми.



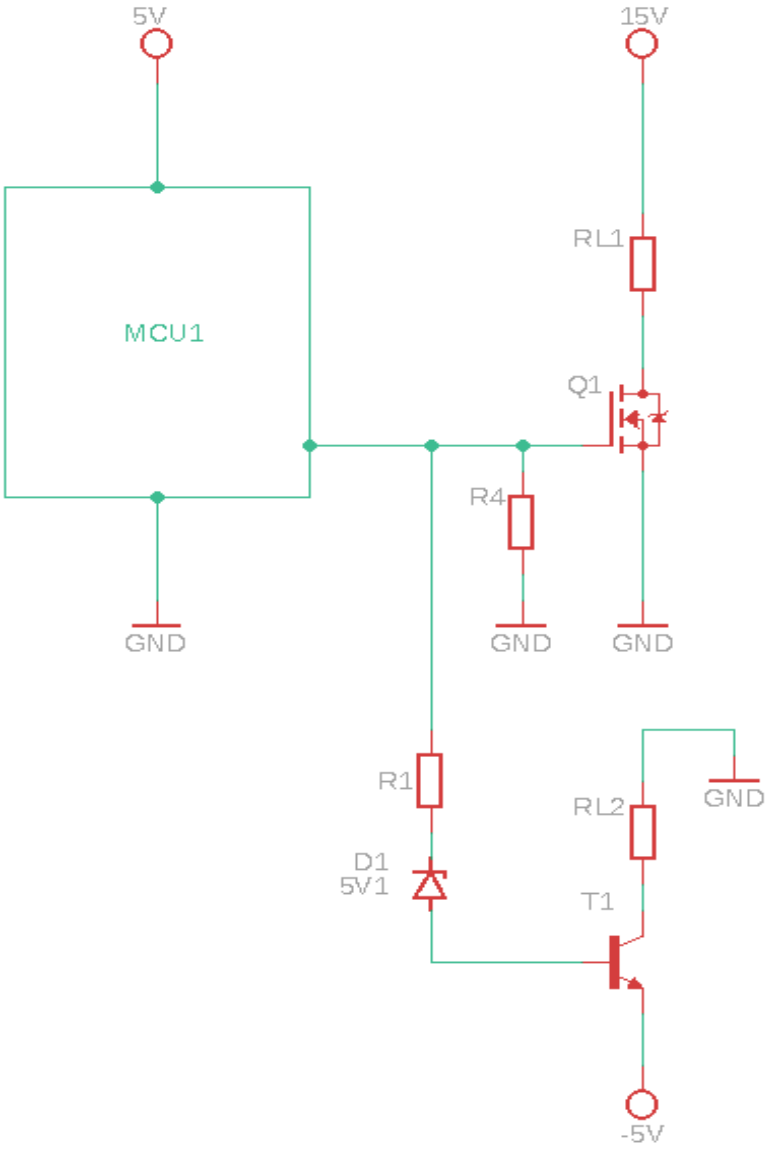
Схемотехника на I/O стъпала

Когато T1 е запушен, гейтът на Q1 е свързан към 15 V през R1 и Q1 е запушен. Когато T2 е отпушен гейтът е на напрежение по-ниско от 15V (делител R1/R2) и създалата се потенциална разлика отпушва Q1.



Схемотехника на I/O стъпала

Управление на два товара едновременно, захранени с положителни и отрицателни напрежения. Ценеровият диод D1 въвежда праг на сработване на T1. Когато на извода има 0V, T1 е запушен. Когато на извода има напрежение > (Vf_D1 + Vbe_T1), T1 ще се отпуши.



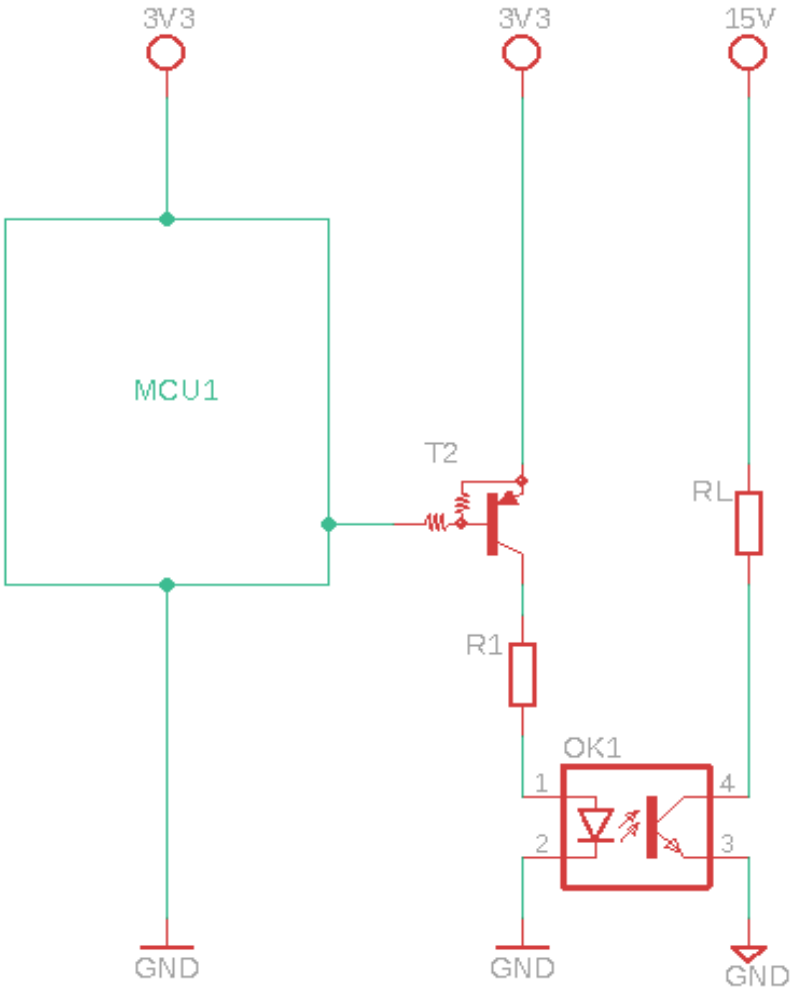
Резисторът R1 се изчислява:

R1 = V_R1 / If_D1 =

= [5V - (-5V) - Vf_D1 – Vbe_T1] / If_D1

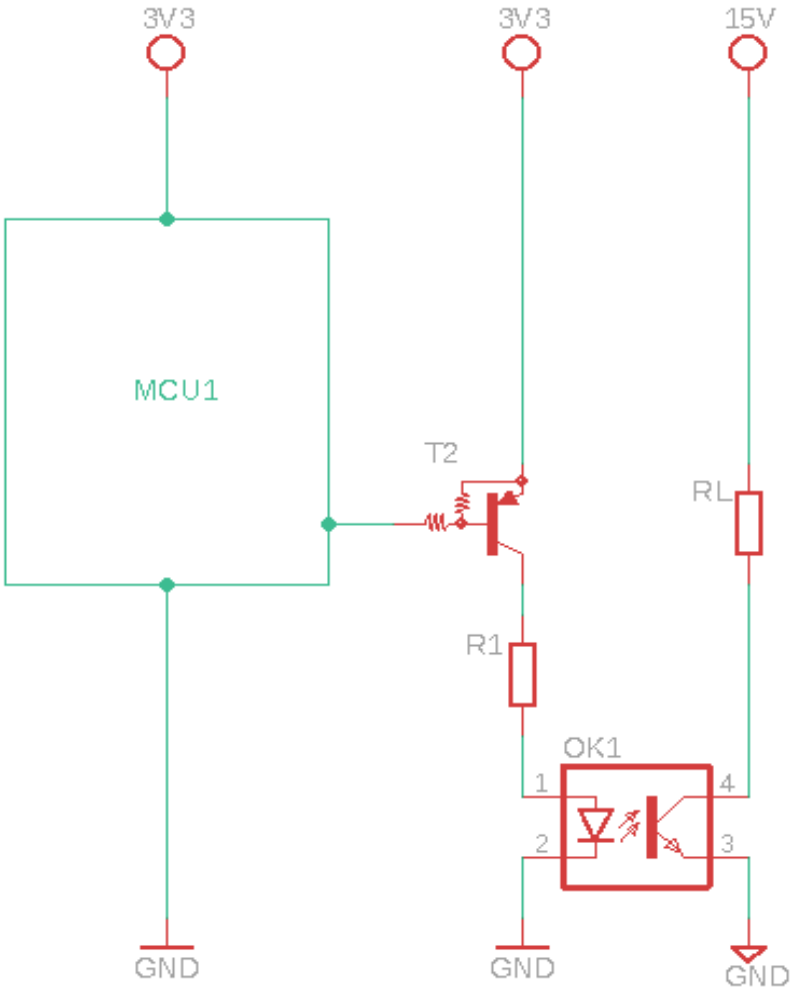
Схемотехника на I/O стъпала

Неинвертираща схема с галванично разделяне. Транзисторът T2 се нарича цифров транзистор (digital transistor, prebiased transistor). R1 трябва да пусне достатъчно ток през диода на оптрона OK1, така че коеф. на предаване по ток да е > (50 ÷ 100) %.



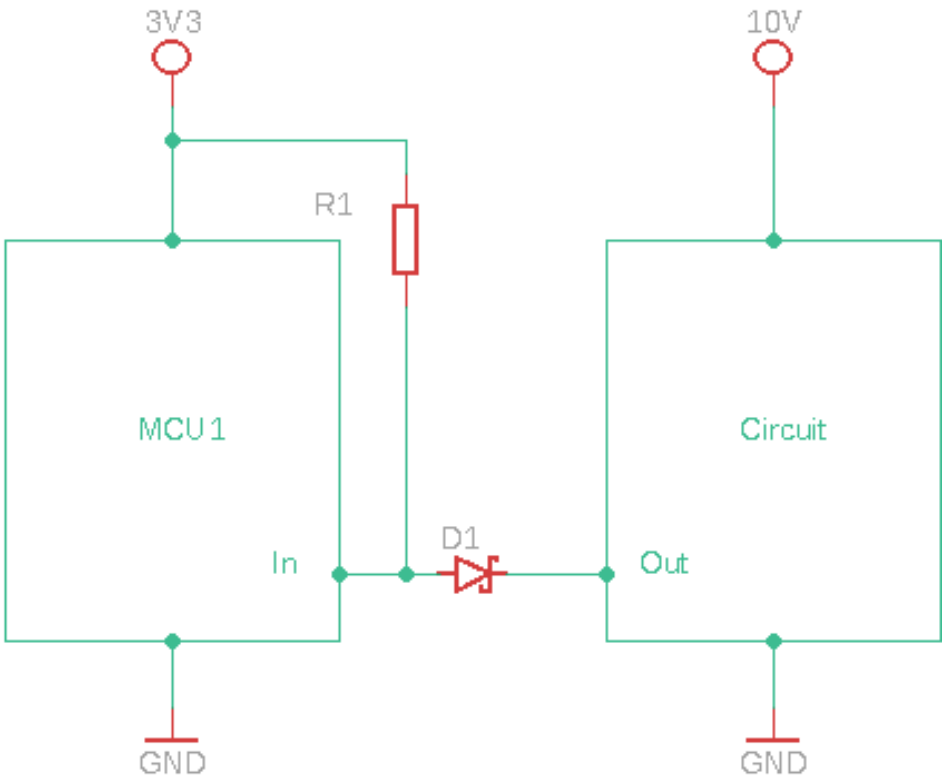
Схемотехника на I/O стъпала

Неинвертираща схема с галванично разделяне. Транзисторът T2 се нарича цифров транзистор (digital transistor, prebiased transistor). R1 трябва да пусне достатъчно ток през диода на оптрона OK1, така че коеф. на предаване по ток да е > (50 ÷ 100) %.



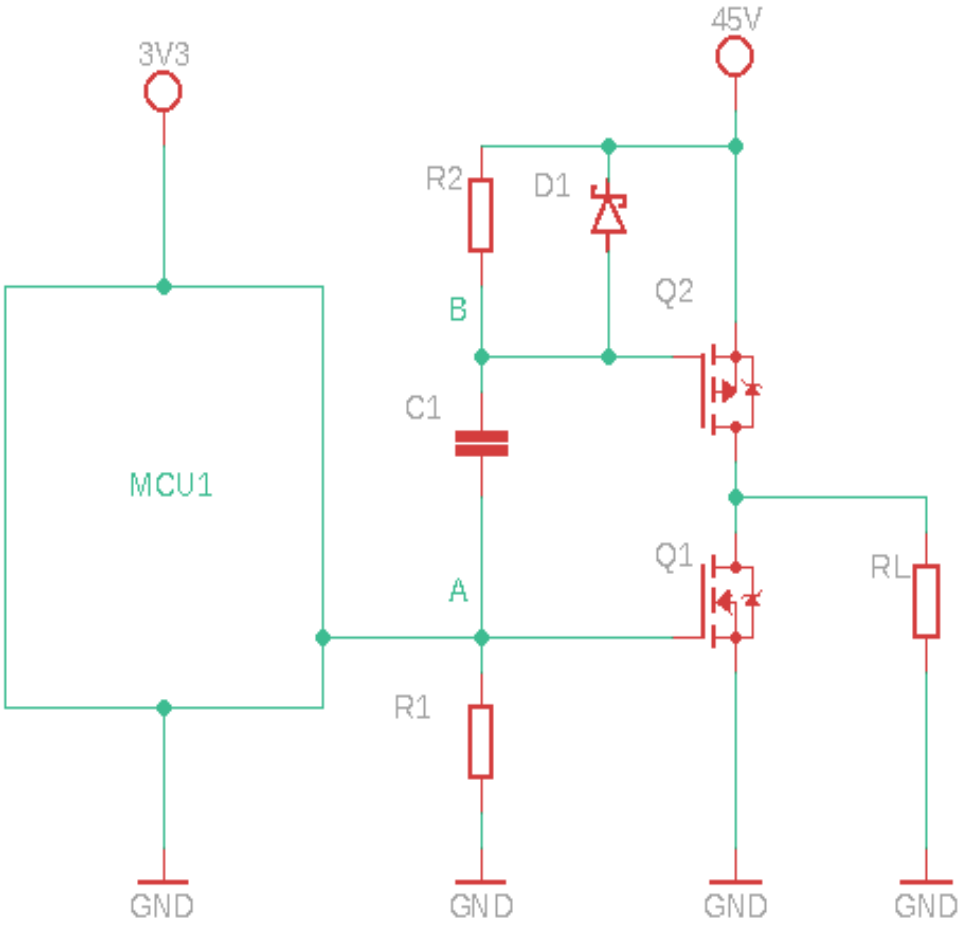
Схемотехника на I/O стъпала

Транслиране с диод, неинвертираща схема. Когато схемата Circuit подаде логическа 0 на изхода си, диодът D1 ще се отпуши през R1. D1 се препоръчва да е Шотки, защото Vf = 0.2 ÷ 0.4V, а V_InputLow за 3.3-волтова CMOS логика е 0.8 V. Когато Circuit установи логическа единица (в случая 10V), диодът D1 се запушва и изводът In остава свързан към 3V3 през R1. Предаването на данни става от Circuit към MCU1.



Схемотехника на I/O стъпала

Транслиране с кондензатор, с противотактен изход [2]. Схемата инвертира. C1 се зарежда през R2. Когато на изхода на MCU1 се подават правоъгълни импулси в т. A, импулси със същата амплитуда ще се виждат и в т. B, но отместени с постояннотокова съставка, равна на високото напрежение (в случая 45V). **Схемата работи само в динамичен режим.** Диодът D1 разрежда C1 до Vhigh (45V), когато в т.A са подадени 3.3V.



Схемотехника на I/O стъпала

Двупосочно транслиране [3].

Изходи на MCU1 и MCU2 = отворен дрейн.

==========

IN/OUT(MCU1) = 0V→
V_GS = 3.3V →
Q1 отпушва, дрейнът му = 0V→
IN/OUT(MCU2) = 0V

==========

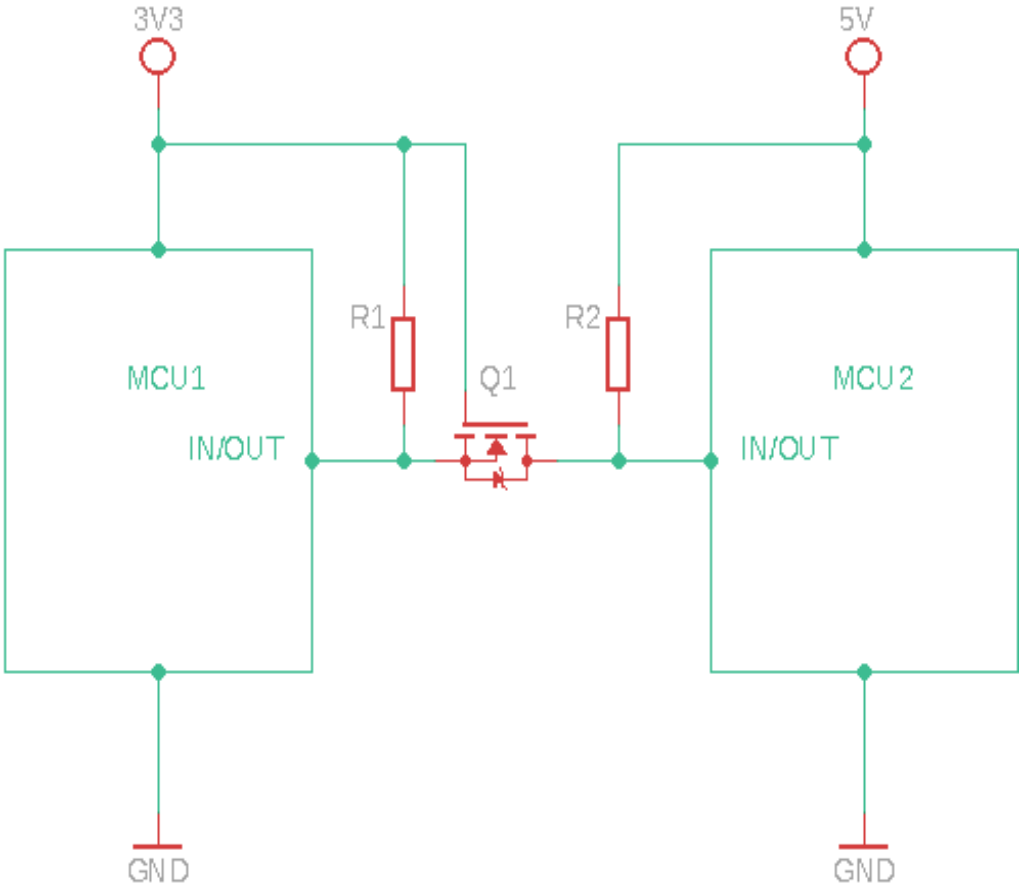
IN/OUT(MCU1) = 3.3V→
V_GS = 0V →
Q1 запушва, дрейнът му = 5V (през R2)→
IN/OUT(MCU2) = 5V

==========

IN/OUT(MCU2) = 0V→
body диода на Q1 отпушва→
IN/OUT(MCU1) = (0.2 ÷ 0.4)V = V_S→
V_GS = 3.3V - (0.2 ÷ 0.4)V→Q1 отпушва →
IN/OUT(MCU1) = 0V (V_S = V_D = 0V)

==========

IN/OUT(MCU2) = 5V→
V_D =5V→
V_S = V_G = **IN/OUT(MCU1) = 3.3V** (през R1, Q1 е запушен, V_GS = 0V)



Входно-изходен модул (GPIO) с общо предназначение

Управлението на входно/изходните изводи често се разделя на групи от по 8, 16, 24, 32 извода. Микропроцесорът извършва четене и установяване (в 0/1) на изводите на интегралната схема чрез специален модул, който се нарича **входно-изходен порт с общо предназначение** (от англ. ез. **G**eneral **P**urpose **I**nput-**O**utput module).

Съществуват различни наименования на GPIO модулите. Например номерацията им може да е с букви (PORTA, PORTB, PORTC ...) или с цифри (PORT0, PORT1, PORT2 ...). Имената на отделните изводи се свързват с имената на портовете + индекс след името на порта. Например извод 1 от PORTA може да се нарече A.1. Извод 4 от порт 2 може да се нарече PIO2.4, и т.н.

Входно-изходен модул (GPIO) с общо предназначение

GPIO модулите представляват съвкупност от регистри, които конфигурират входно-изходните стъпала, свързани към всички изводи в един **микроконтролер**. Всеки регистър си има собствен, уникален адрес. Регистрите от един модул се разполагат на последователни адреси. Възможно е да има повече от един GPIO модул. Тогава към имената на регистрите се добавя индекс.

Пример – GPIO модулите на MSP430FR6989 са 10 броя. Регистрите, от които са изградени са еднотипни и вършат една и съща работа:

*входните регистри се означават P1IN, P2IN, P3IN … P10IN

*изходните P1OUT, P2OUT, P3OUT ... P10OUT

*и т.н.

Входно-изходен модул (GPIO) с общо предназначение

GPIO модулите представляват съвкупност от регистри, които конфигурират входно-изходните стъпала, свързани към всички изводи в един **микроконтролер**. Всеки регистър си има собствен, уникален адрес. Регистрите от един модул се разполагат на последователни адреси. Възможно е да има повече от един GPIO модул. Тогава към имената на регистрите се добавя индекс.

Пример – GPIO модулите на MSP430FR6989 са 10 броя. Регистрите, от които са изградени са еднотипни и вършат една и съща работа:

*входните регистри се означават P1IN, P2IN, P3IN … P10IN

*изходните P1OUT, P2OUT, P3OUT ... P10OUT

*и т.н.

Входно-изходен модул (GPIO) с общо предназначение

За четенето, записа и управлението на даден извод от микроконтролера се използват **регистри със специално предназначение**, при които всеки бит управлява (конфигурира) една функция.

Примерни специални регистри са:

*входен
*изходен
*регистър за посока
*регистър с флагове за прекъсвания
*регистър за разрешаване на прекъсвания
*регистър за вида на прекъсването
*регистър за избор на ниво
*регистър за избор на фронт
*регистър за разрешаване на издърпващи резистори
*регистър за избор на издършващ резистор
*регистър за избор на функция на извода

Входно-изходен модул (GPIO) с общо предназначение

Входен регистър (input register) – използва се за временно съхраняване на логическите състояния на изводите от един порт, когато е конфигуриран като входен.

Изходен регистър (output register) - използва се за установяване на логическите състояния на изводите от един порт, когато е конфигуриран като изходен.

Тези два регистъра са онагледени на фигурите в следващия слайд. Вижда се, че това са буферни регистри. При директно свързване на съответните изводи логическите нива от изходния регистър на MCU1 ще се прехвърлят във всеки един бит (D-тригер) на входния регистър на MCU2. Всяка клетка от паралелните регистри е **D-тригер**.

Входно-изходен модул (GPIO) с общо предназначение

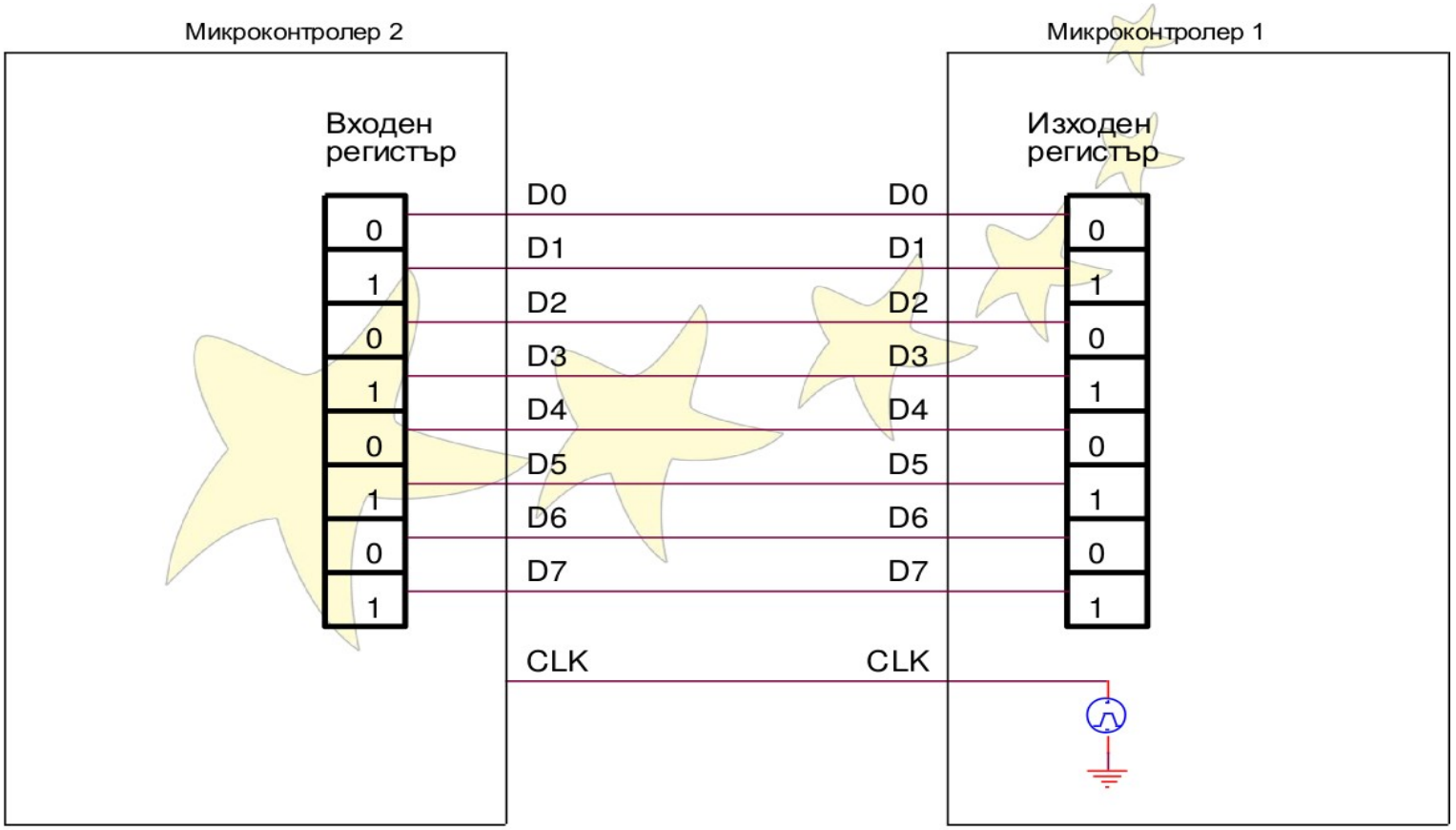
За четенето, записа и управлението на даден извод от микроконтролера се използват **регистри със специално предназначение**, при които всеки бит управлява (конфигурира) една функция.

Примерни специални регистри са:

*входен регистър
*изходен регистър
*регистър за посока
*регистър с флагове за прекъсвания
*регистър за разрешаване на прекъсвания
*регистър за вида на прекъсването
*регистър за избор на ниво
*регистър за избор на фронт
*регистър за разрешаване на издърпващи резистори
*регистър за избор на издършващ резистор
*регистър за избор на функция на извода

*регистър за избор на товароносимост *регистър за избор на стръмност

Входно-изходен модул (GPIO) с общо предназначение



Входно-изходен модул (GPIO) с общо предназначение

Регистър за посока (direction register) – всеки един бит от него отговаря на даден извод и определя дали той ще е вход или изход.

Регистър с флагове за прекъсвания (interrupt flag register) – всеки бит отразява дали на даден извод е настъпило събитие (например дали е детектиран фронт или дали логическото му състояние се е променило).

Регистър за разрешаване на прекъсванията (interrupt enable register) – всеки бит от него разрешава или забранява дадено прекъсване и когато то се появи (в регистъра с флаговете) микропроцесорът спира изпълнението на главната програма, след което обслужва специална подпрограма, наречена *прекъсване*.

Входно-изходен модул (GPIO) с общо предназначение

Регистър за вида на прекъсване – указва дали прекъсването ще се регистрира по фронт или по ниво.

Регистър за избор на фронт (edge select register) – указва по кой фронт (нарастващ или падащ) да се регистрира прекъсване в регистъра с флаговете.

Регистър за избор на ниво (level select register) – указва по кое логическо ниво (нула или единица) да се регистрира прекъсване в регистъра с флаговете.

Регистър за разрешаване на издърпващи резистори (pull resistor enable) – всеки един бит отговаря за включването на издърпващ резистор на съответния извод.

Входно-изходен модул (GPIO) с общо предназначение

Регистър за избор на издърпващ резистор – избира дали съответния издърпващ резистор да бъде към захранване (pull-up) или към маса (pull-down).

Регистър за избор на функцията на извода (gpio mode / gpio select / gpio multiplex register) – избира дали съответния извод да е свързан към GPIO модул или към друг вътрешен модул (напр. АЦП, таймер, компаратор и т.н.). Повечето микроконтролери стартират с всички изводи, свързани към GPIO модули.

Регистър за избор на товароносимост – конфигурира токът на изходните стъпала. По-мощните изходи могат да отдават/приемат по-големи токове, но загубната статична мощност е голяма. По-маломощните изходи имат по-малка статична мощност, но могат да отдават/приемат по-малки токове.

Входно-изходен модул (GPIO) с общо предназначение

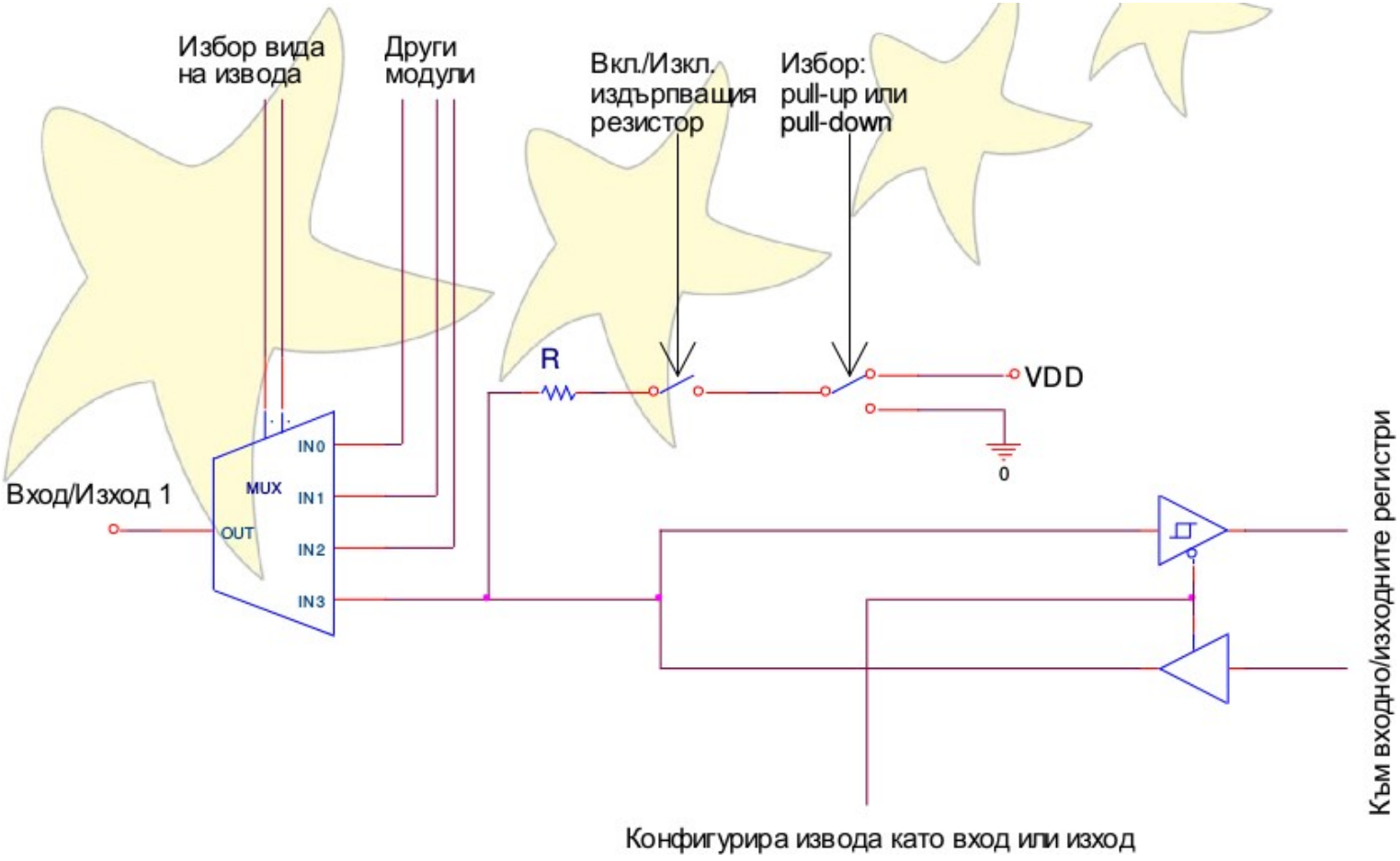
Регистър за избор на стръмност – конфигурира стръмността на фронтовете за съответните изводи.

Много стръмни фронтове в някои мощни схеми могат да предизвикат осцилации (звънене), които да смущават останалата част от системата. В тези случай е добре да се намали КПД, за да се намалят шумовете.

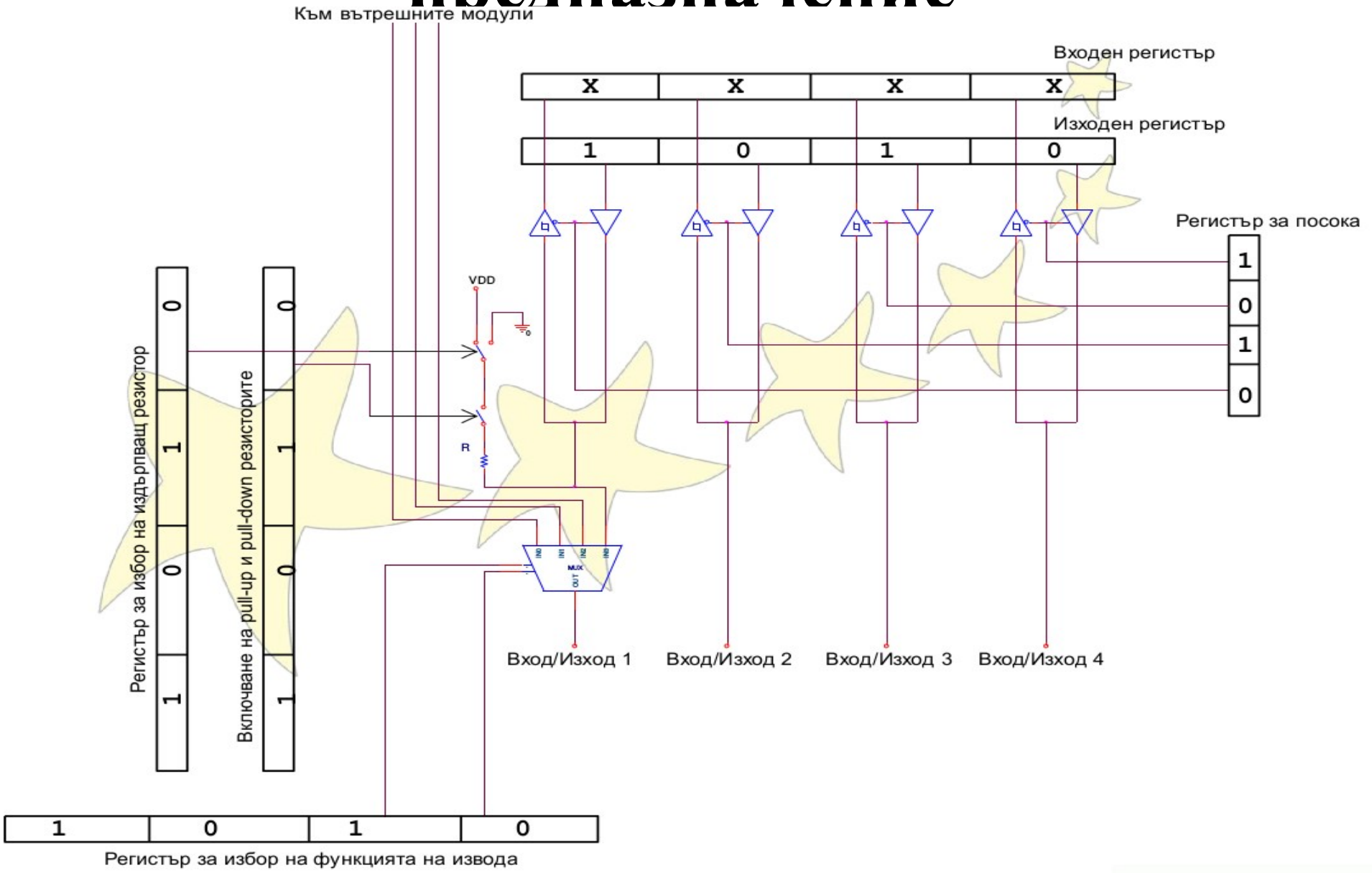
Пример – микроконтролерът LM3S9B92 има регистър GPIOSLR, който позволява стръмността на фронтовете за 8-милиамперовия обхват на GPIO-тата да бъде намалена с до 30 % (бит SRL = 1) или да бъде непроменена (бит SRL = 0).

Входно-изходен модул (GPIO) с общо предназначение

Показана е опростена принципна схема на 4 входно-изходни извода. На фигурата по-долу са дадени само имената на сигналите без регистрите.



Входно-изходен модул (GPIO) с общо предназначение

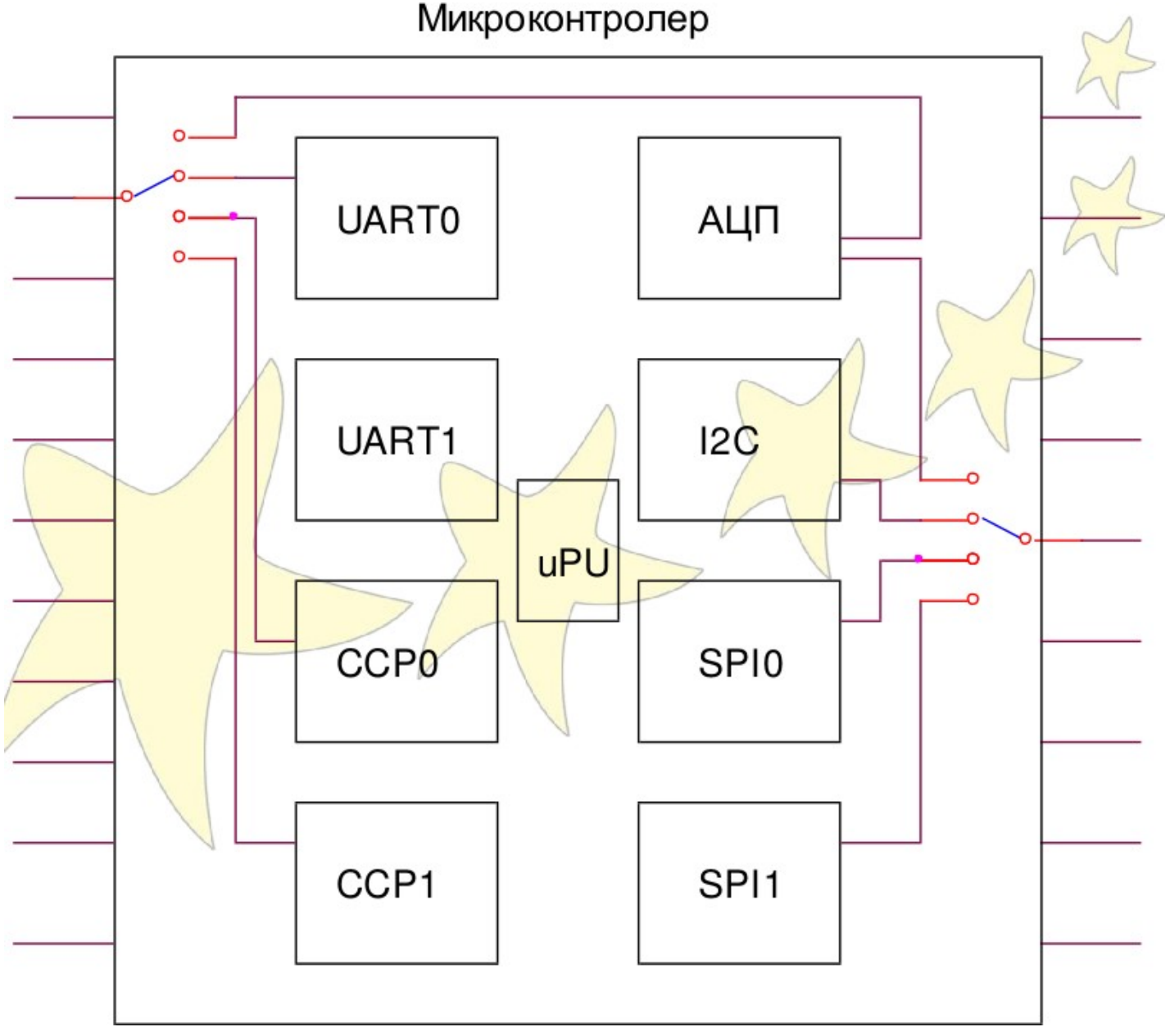


Входно-изходен модул (GPIO) с общо предназначение

Мултиплексиране на изводите – микроконтролерите включват в структурата си много и разнообразни модули, които не винаги се използват в дадено приложение. Затова от гледна точка на цената е по-изгодно да се използва корпус с по-малък брой изводи, които да имат две или повече функции.

Като недостатък на този подход може да се посочи **невъзможността** за използване на **два модула едновременно**, ако те споделят един и същ извод. На следващия слайд е представена блокова схема на микроконтролер, който използва мултиплексиране на изводите си.

Входно-изходен модул (GPIO) с общо предназначение



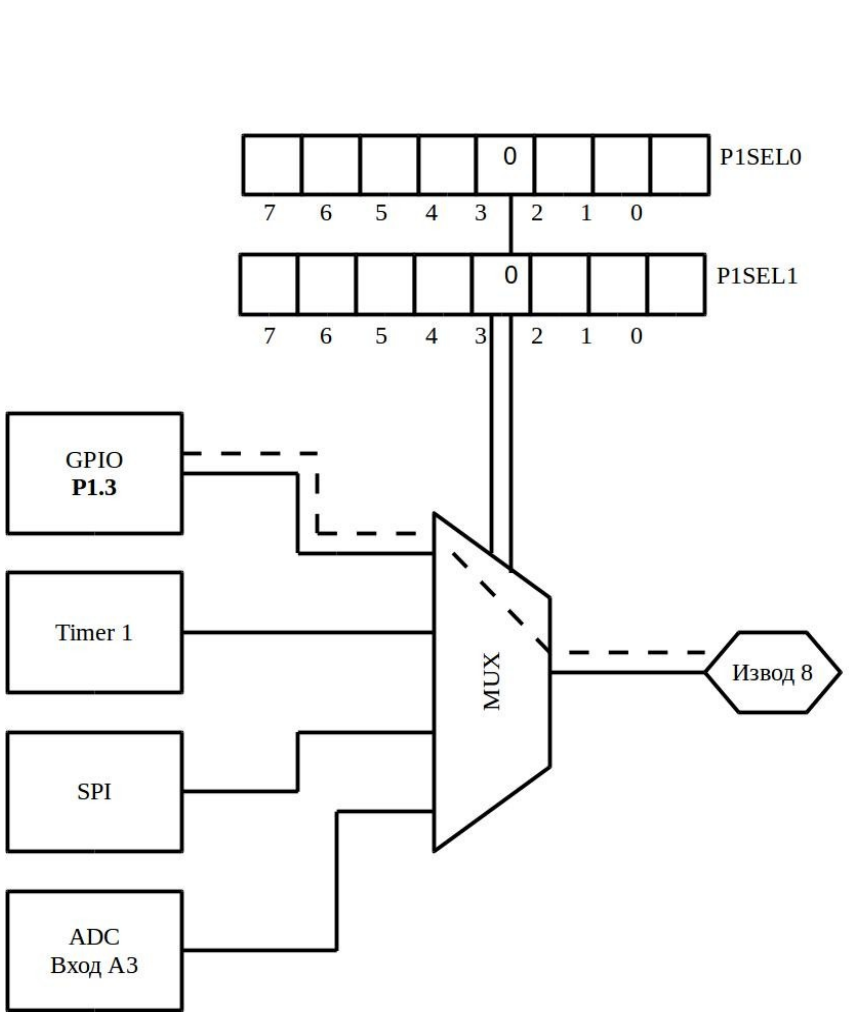
Входно-изходен модул (GPIO) с общо предназначение

Пример – Sitara AM3353 има регистър CONF_GPIO_x, в който се конфигурират входно/изходните стъпала и мултиплексорите:

Бит 6 – избор на стръмност на фронтовете
Бит 5 – конфигурира извод x като вход
Бит 4 – избира вид на издърпващ резистор
Бит 3 – разрешава издърпващ резистор
Бит <0 – 3> - избира функция на извода (до 8 модула на един извод).

Входно-изходен модул (GPIO) с общо предназначение

Пример – MSP430FR6989 има регистри PxSEL0 и PxSEL1, в които се конфигурира мултиплексора



Номер на двойката регистри PxSELx.
Съвпада с номера на GPIO порта.

Номер на регистър от двойката PxSELx.
Може да е 0 или 1.

Позиция на бита в PxSELx регистрите.
Съвпада с позицията на бита от GPIO регистрите.

P1SEL1.**3**	**P1**SEL0.**3**	Функция	
0	0	GPIO	(основна)
0	1	Timer 1	(първична)
1	0	SPI	(вторична)
1	1	ADC	(третична)

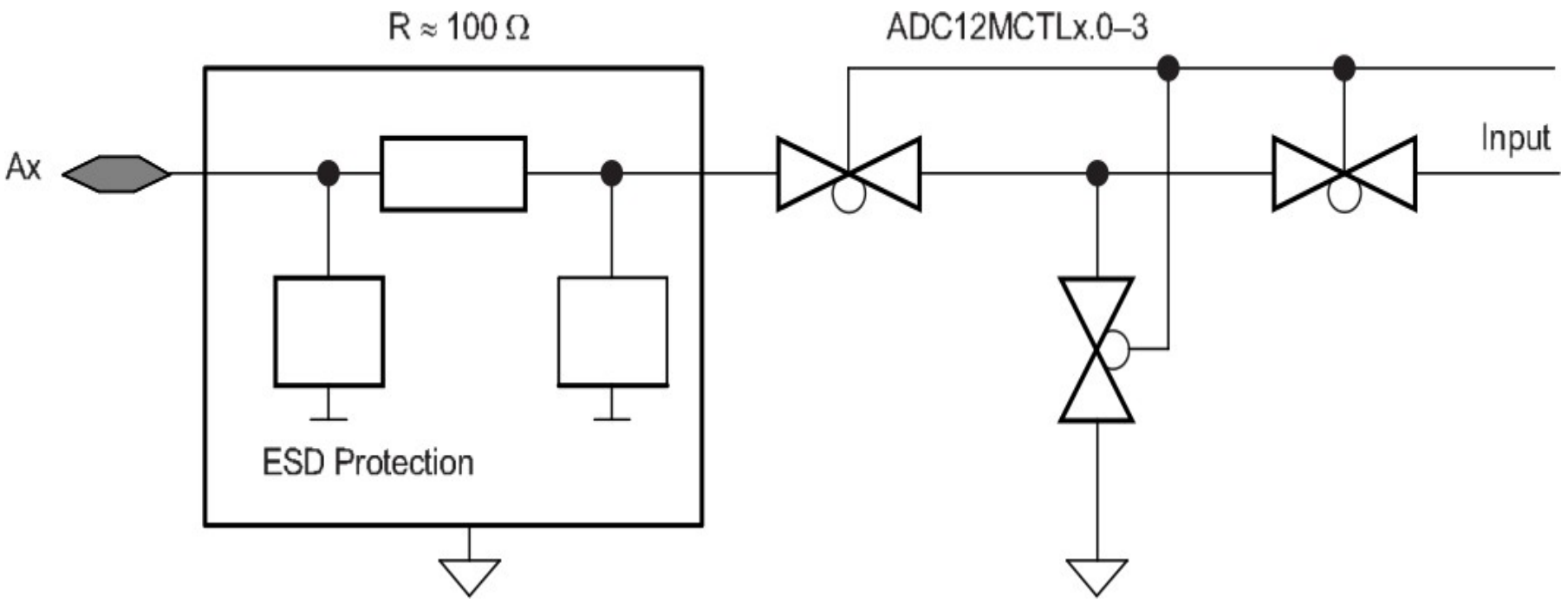
Входно-изходен модул (GPIO) с общо предназначение

Изводи, свързани към аналогови схеми (АЦП, ЦАП, аналогови компаратори, програмируеми усилватели и т.н.) се мултиплексират с **аналогови ключове**.

Някои микроконтролери свързват неизползваните аналогови входове на маса, за да намалят шумовете между отделните канали.

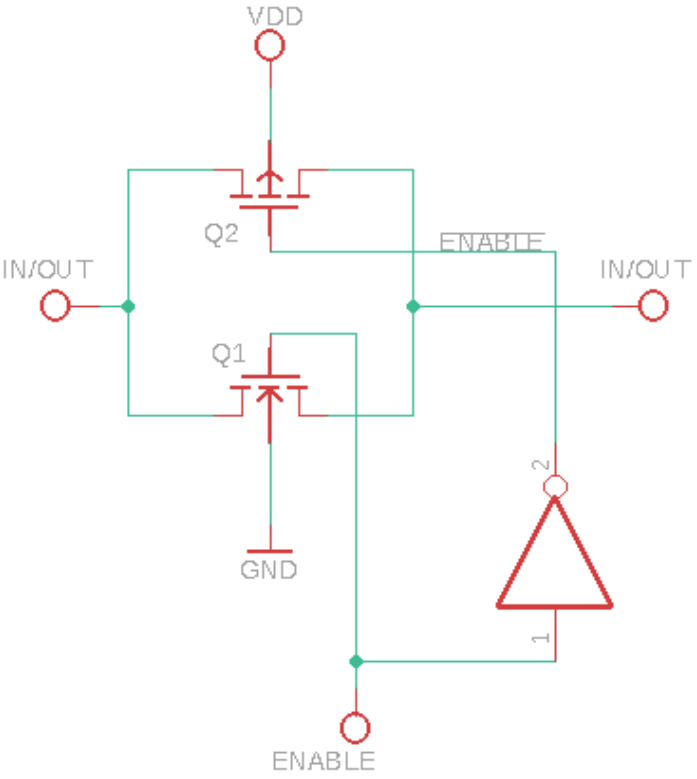
Входно-изходен модул (GPIO) с общо предназначение

Пример – MSP430FR6989 има следната принципна схема на аналоговия си мултиплексор:



Входно-изходен модул (GPIO) с общо предназначение

Аналоговият CMOS ключ има принципната схема показана вдясно. Той има два терминала за захранване (VDD, GND), един за включване/изключване (ENABLE), и два входно/изходни терминала (IN/OUT). Последните са **взаимозаменяеми**.



Два транзистора са небходими, за да се преодолее недостатъкът от праговите напрежения на MOS:

*NMOS не може да провежда напрежения, близки до логическа 1

*PMOS не може да провежда напрежения, близки до логическа 0.

Оперативна памет SRAM

Оперативната памет на микроконтролерите е със свободен достъп (**R**andom **A**ccess **M**emory) и е енергозависима.

В повечето случай се съхраняват променливите на програмата.

Възможно е и самата програма да се помести в RAM (по-рядко се прави), ако при стартирането на системата някой я копира от ROM в RAM.

Вградената в микроконтролери RAM е от вида **SRAM** (**S**tatic **R**andom **A**ccess **M**emory). На следващия слайд е показана схемотехниката за запаметяване на 1 бит информация.

Оперативна памет SRAM

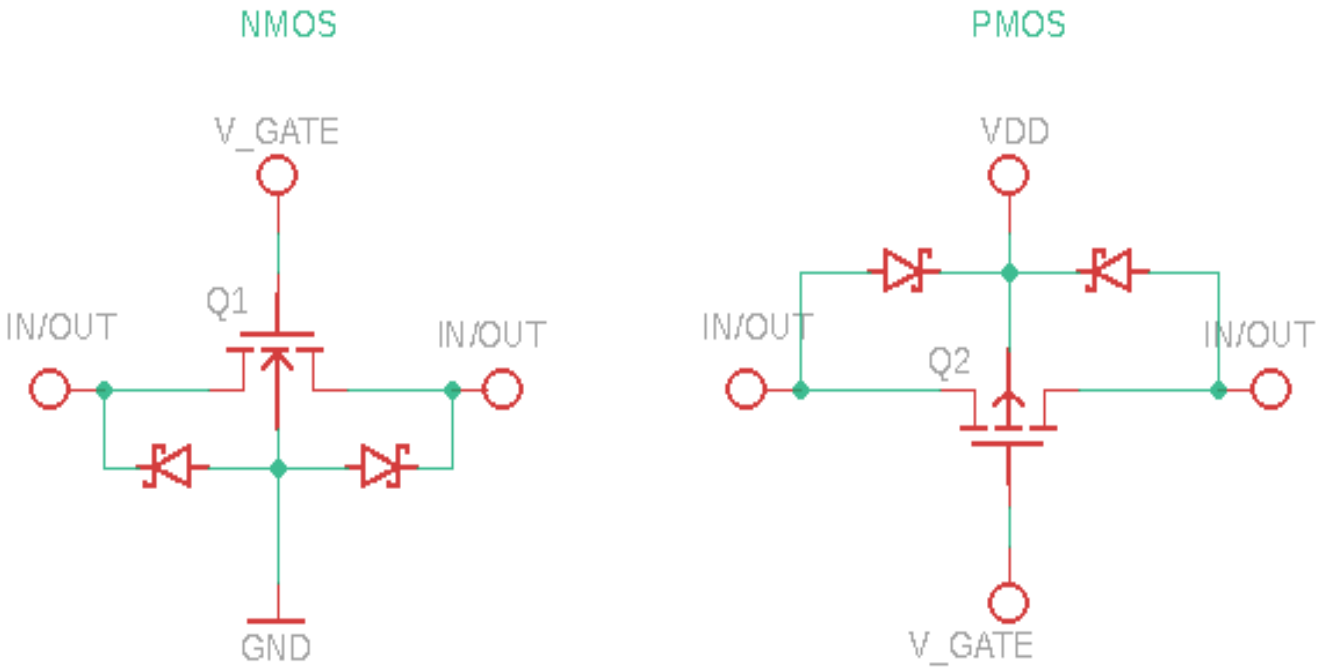
!!!ВНИМАНИЕ!!! - в CMOS технологията дрейн и сорс областите на MOSFET-ите се правят симетрични (за разлика от мощните MOSFET), а подложките се свързват:

*към маса за NMOS
*към захранване за PMOS

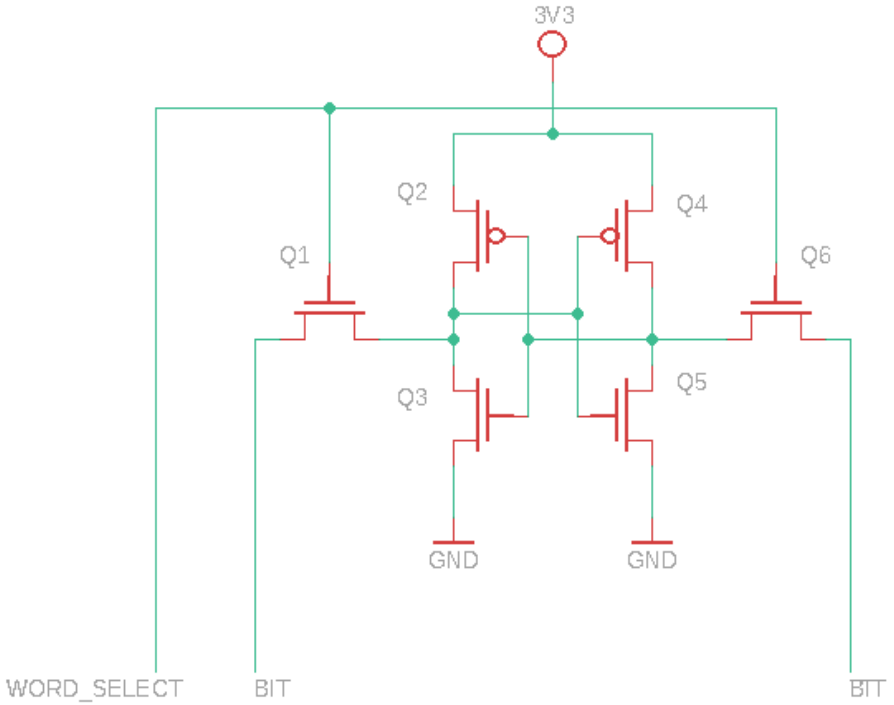
Това означава, че електродите на дрейна и сорса са взаимозаменяеми и **ток** може да тече от **дрейн към сорс** и от **сорс към дрейн**. Тоест на NMOS гейта се подава захранване, за да се отпуши, а на PMOS гейта се подава маса. За да се запуши NMOS на гейта се подава маса, а на PMOS – захранване.

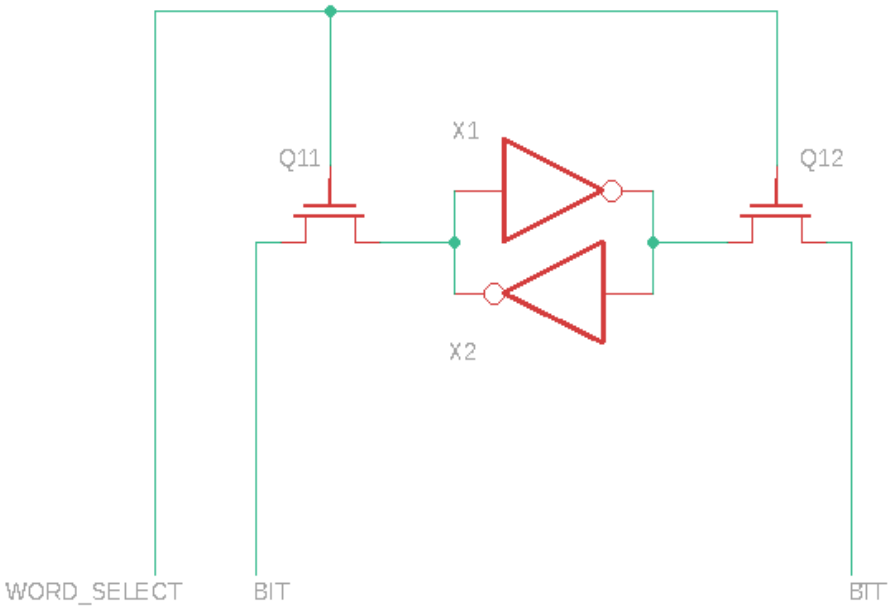
Оперативна памет SRAM

Еквивалентни схеми на NMOS и PMOS в CMOS интегрална схема. Транзисторите се използват като двупосочни ключове. При цифрови схеми се използва само един транзистор, защото е допустимо да се внася грешка от праговото напрежение Vth0 на MOS транзисторите. Символите на NMOS и PMOS в интегрално изпълнение често не показват боди електрода.



Оперативна памет SRAM





Оперативна памет SRAM

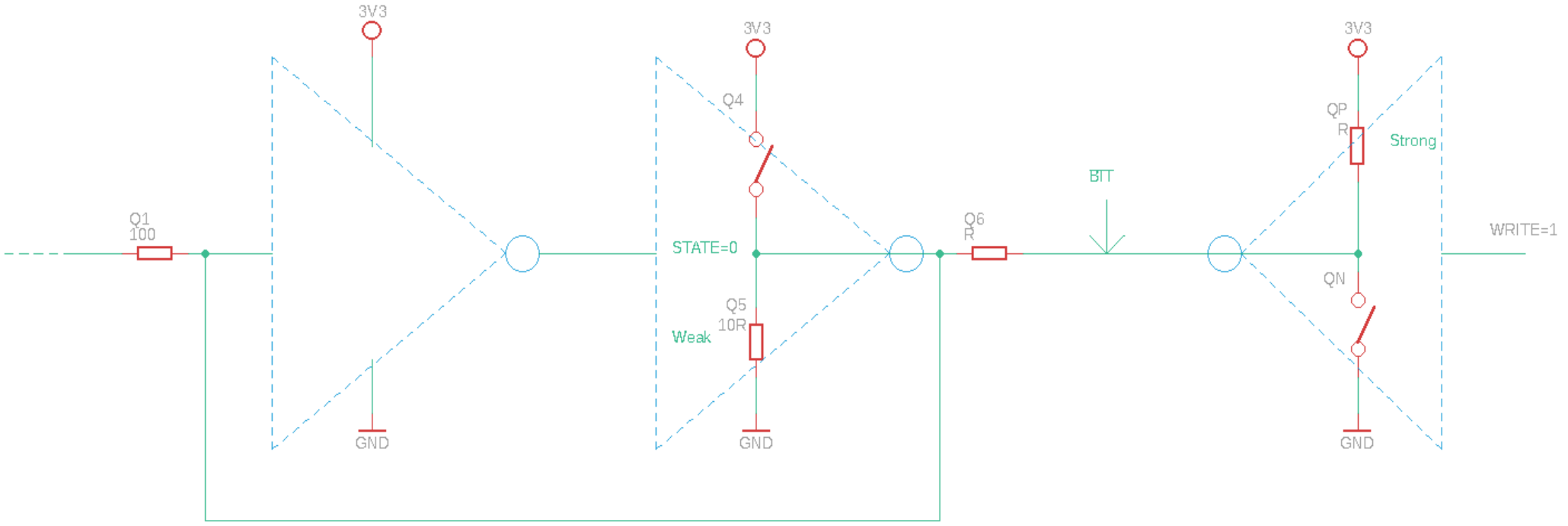
Всеки бит се запаметява от тригер. Тригерът е съставен от 4 транзистора (Q2, Q3, Q4 и Q5) и се достъпва с 2 (Q1 и Q6). Това е т.нар. **6T клетка.** Тригерът е асинхронен RS с един вход.

При запис на бит, еднакъв с досегашното състояние на клетката, транзисторите на тригера не променят състоянието си.

При запис на бит, различен от досегашното състояние на клетката, транзисторите на тригера променят състоянието си под влиянието на мощните транзистори, които записват този бит. Мощните транзистори имат по-малко съпротивление на канала и могат да отдадат/приемат по-голям ток, който да преобърне тригера.

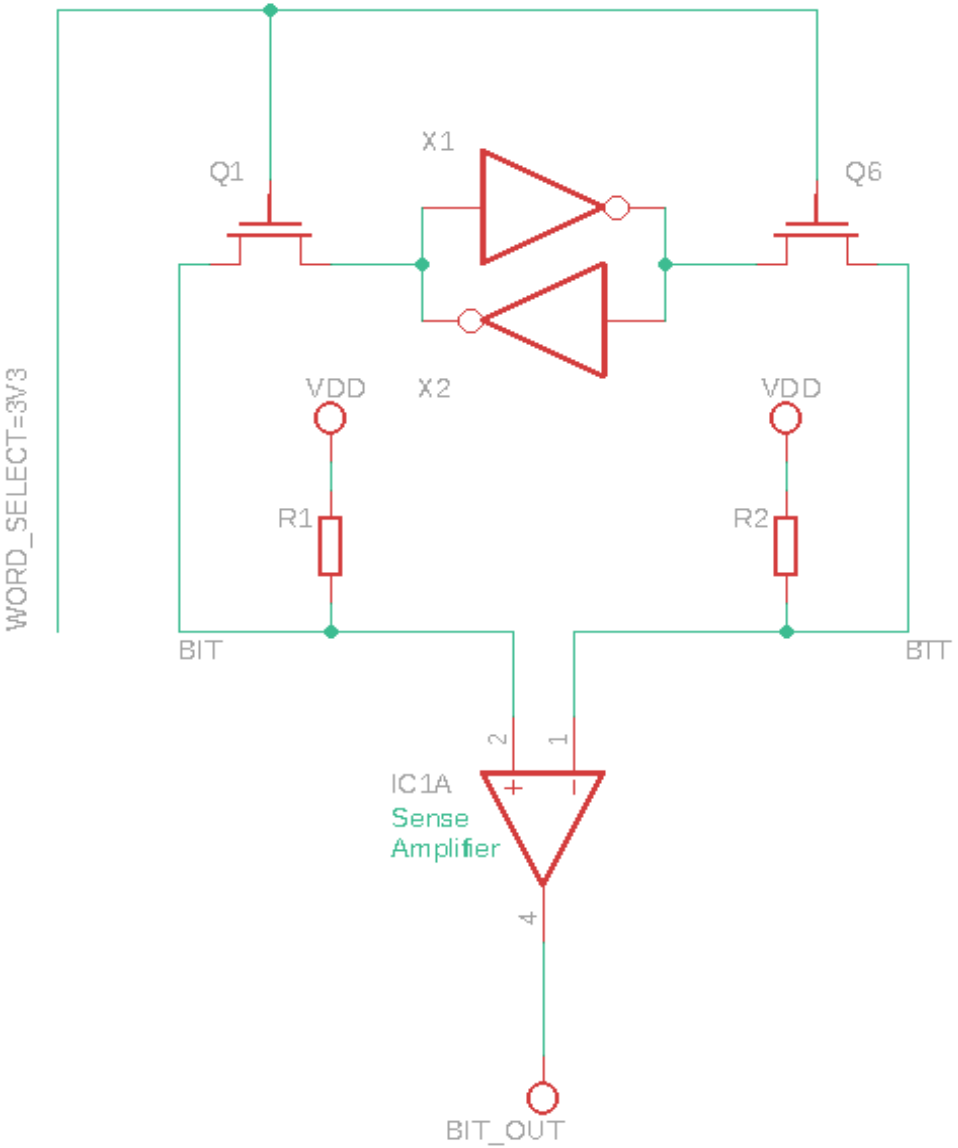
Оперативна памет SRAM

Еквивалентна схема на десния инвертор (Q4, Q5), десния транзистор за достъп Q6 и външен мощен инвертор, който осъществява запис на 1, когато сигналът !BIT е бил в 0.



Оперативна памет SRAM

Два изхода на тригера са необходими, за да бъдат свързани към **компараторна схема** (voltage comparator, sense amplifier) или още срещана като **четящ усилвател** [6]. Тя изработва правилни логически нива, когато клетката се чете (WORD_SELECT=VDD).



Оперативна памет SRAM

Защо е необходим компаратор при положение, че SR-тригерът запаметява логическото състояние, в което е, до безкрай?

Оперативна памет SRAM

Всеки един бит използва 6 транзистора → стремежът е те да са **по-малки** и **повече**.

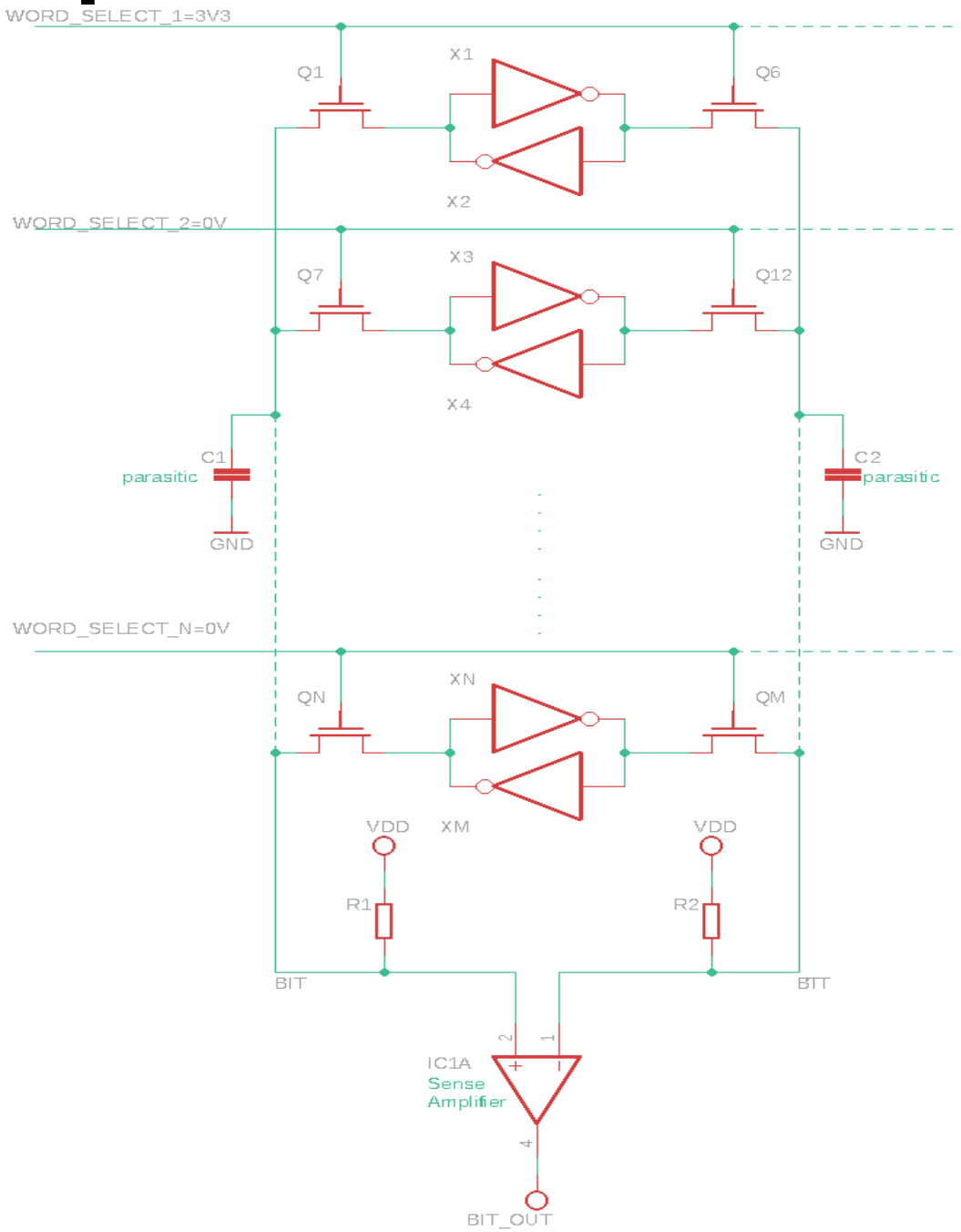
Сигналната линия за битове BIT (още наричана column signal) е свързана паралелно към **много клетки**.

Дори и да не се достъпват тези клетки в дадения момент, те натоварват линията с **паразитен капацитет**.

Когато се активира линията за четене (още наричана row select) този капацитет се **зарежда/разрежда много бавно** през високоомните канали на тригерните транзистори.

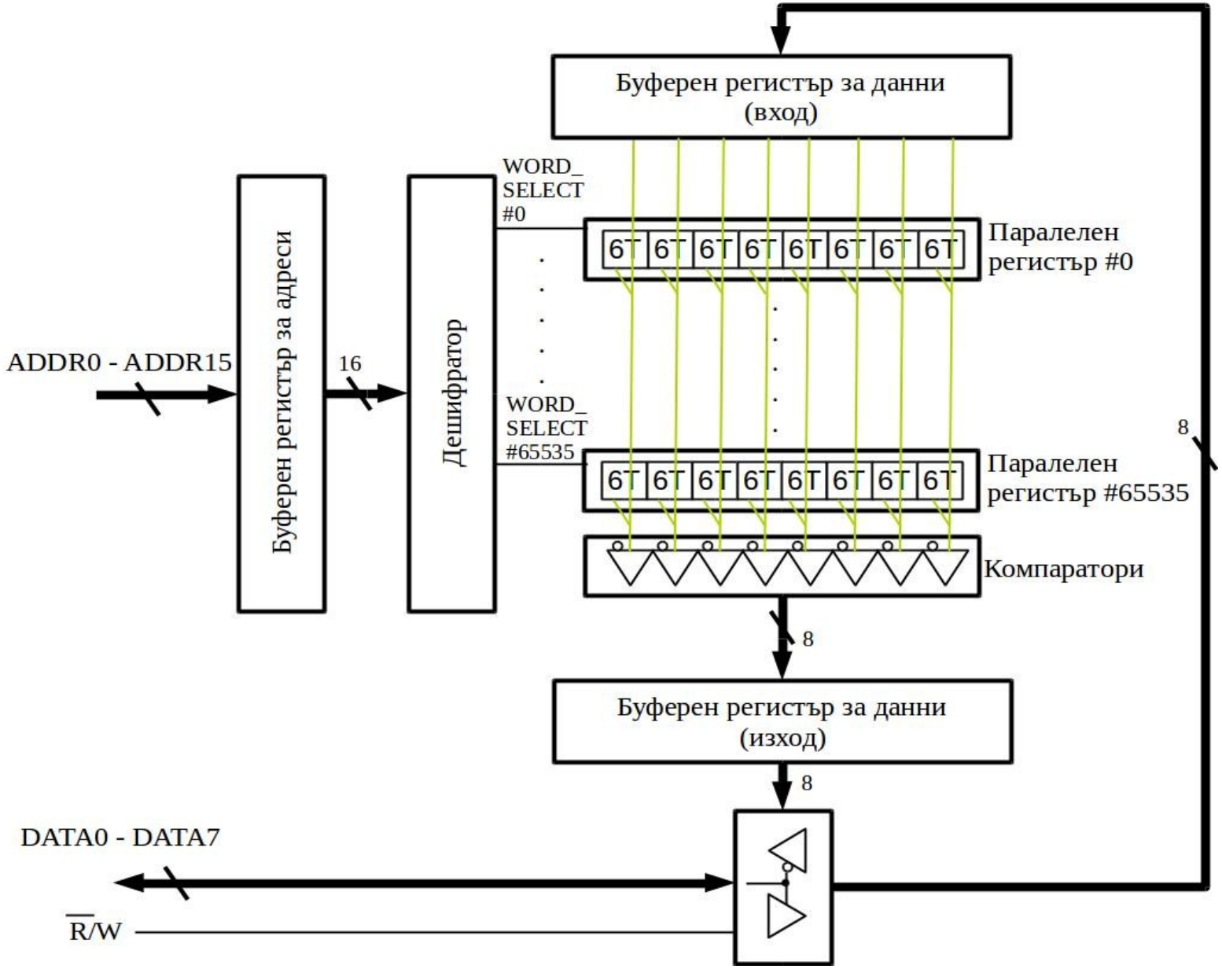
Следователно, добавянето на компаратор ще **ускори четенето във времето** (т.е. не се изчаква пълния заряд/разряд на кондензатора) [4].

Оперативна памет SRAM



Оперативна памет SRAM

Опростена структурна схема на 64-килобайтова SRAM.

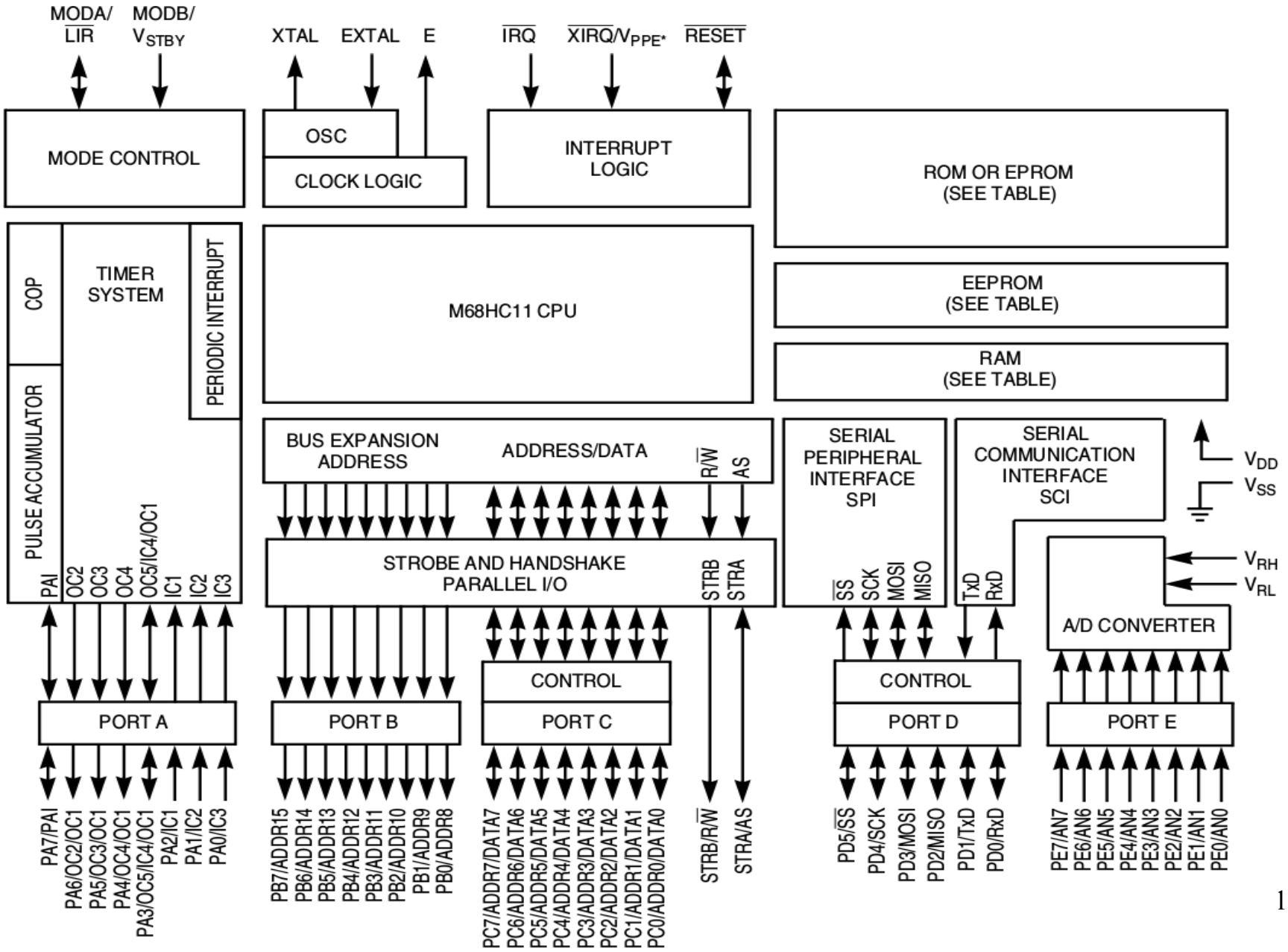


Оперативна памет SRAM

Пример – класическият микроконтролер MC68HC11 може да работи в режим на микропроцесор. В този режим адресната и даннова магистрала на микропроцесора се включват към изводите на чипа.

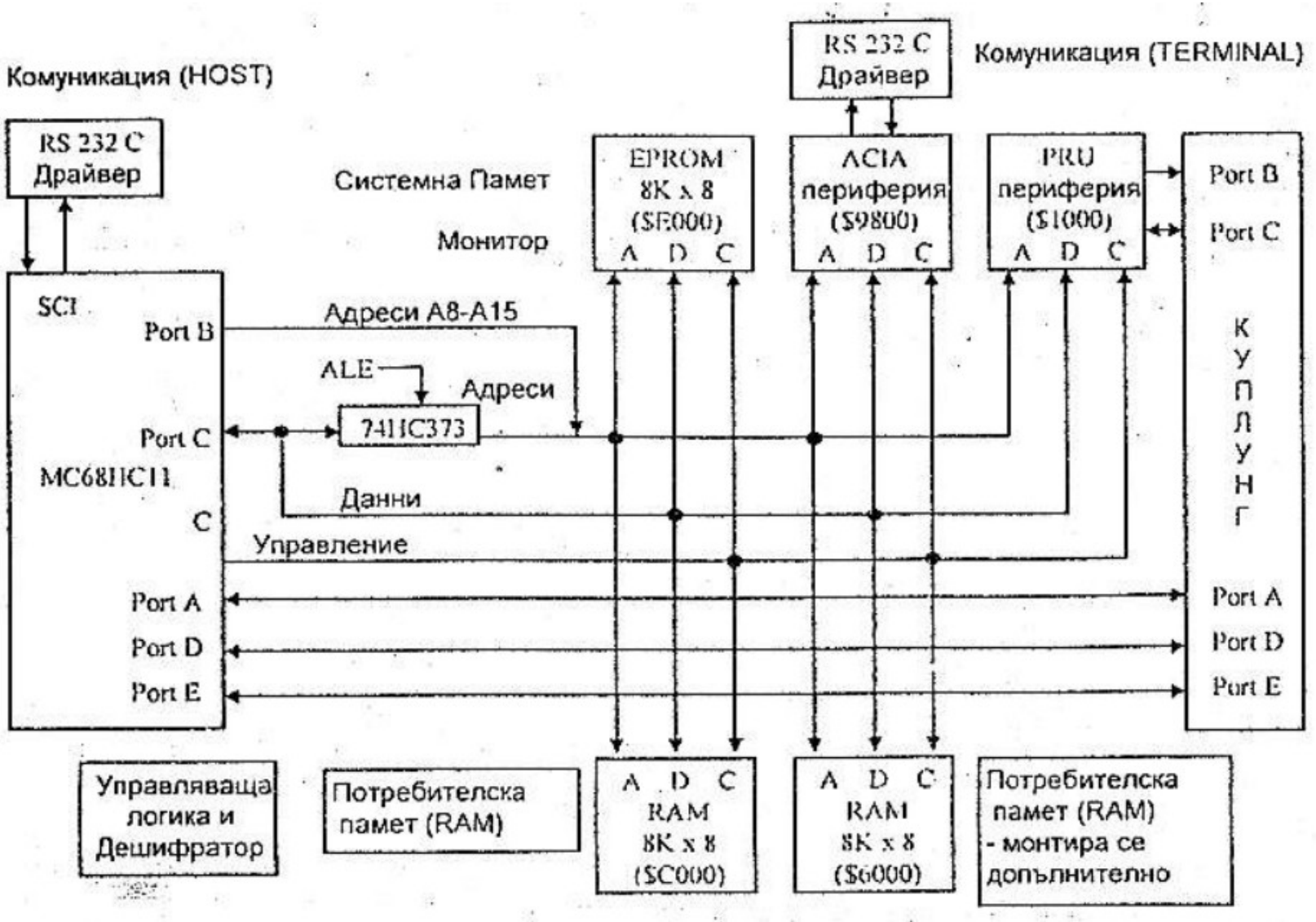
За да се спестят изводи, сигналите на младшата част на адресите (ADDR0 ÷ ADDR7) и сигналите на данните (DATA0 ÷ DATA7) са изведени на едни и същи изводи. Един специален извод (адресен строб, AS) указва дали в дадения момент на тези изводи има адреси или данни. С помощта на външен паралелен регистър трябва да се **мултиплексират във времето** (time multiplexing). Обменът се забавя, заради тази особеност. За регистър може да се използва 74HC373.

Оперативна памет SRAM



Оперативна памет SRAM

Блокова схема на микропроцесорна система с M68HC11 [5].

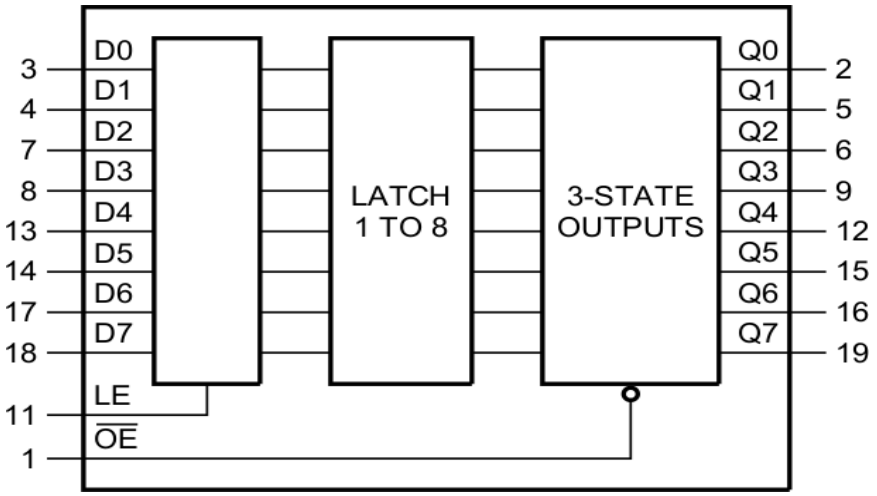


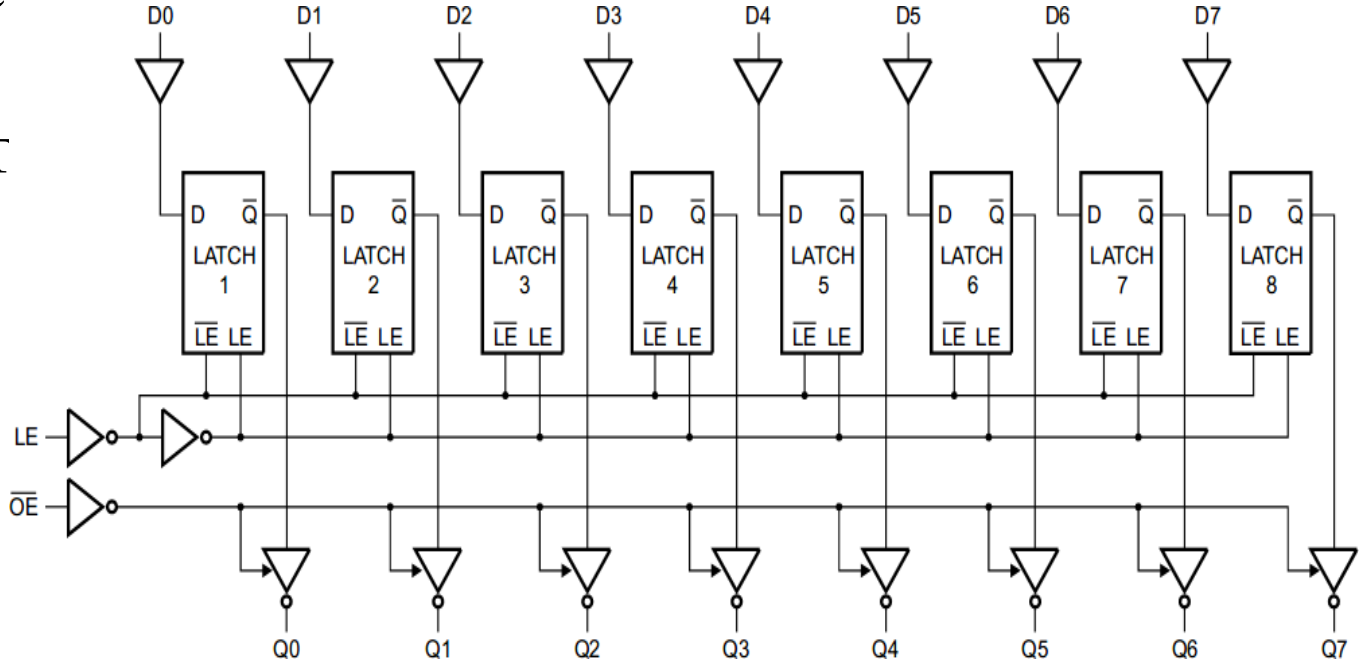
Оперативна памет SRAM

74HC373

==========

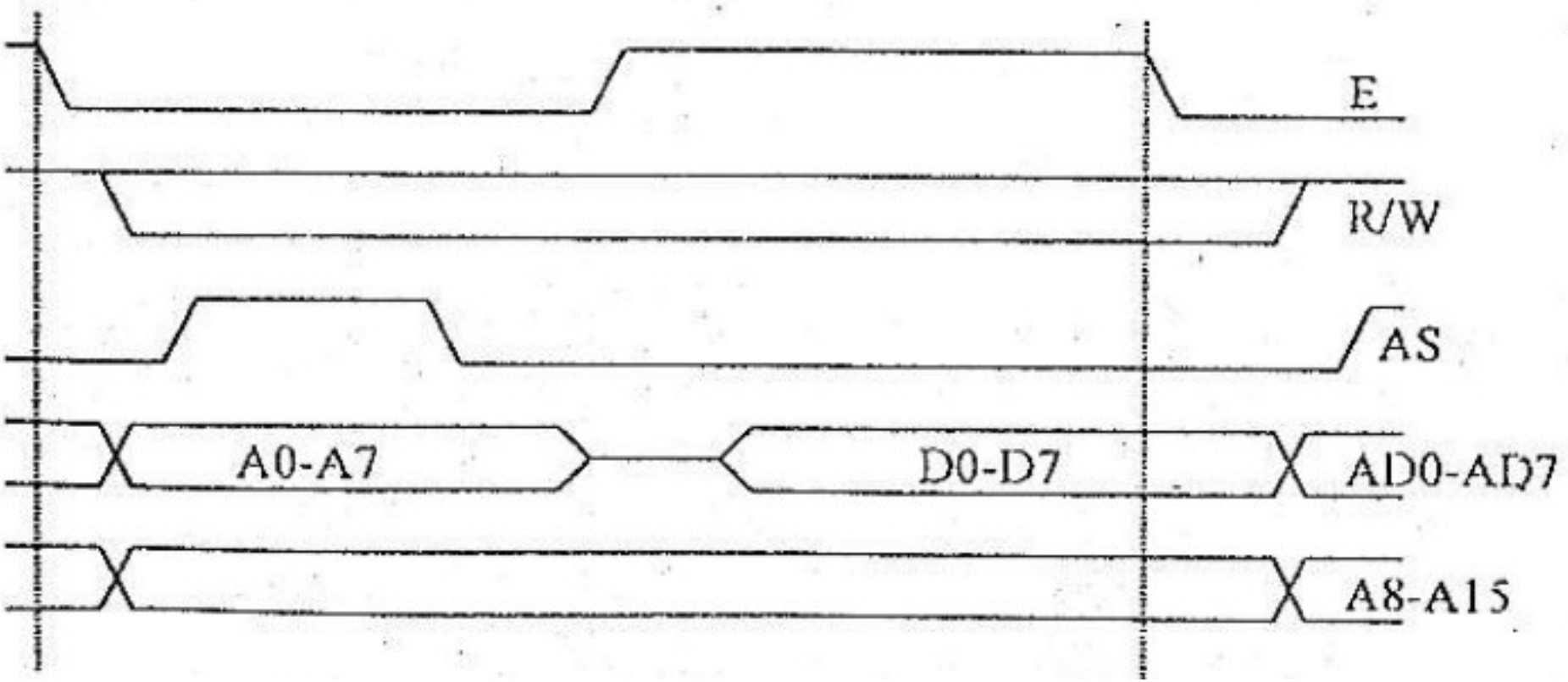
D0 ÷ D7 – входове
Q0 ÷ Q7 – изходи
!OE – разрешаване на изходите, постоянно свързан към маса
LE – запаметяване на числото, подадено на D0 ÷ D7 в D-тригерит





Оперативна памет SRAM

Обмен на данни между микропроцесор HC11 и външна SRAM [5].



Оперативна памет DRAM

В оперативната динамична памет (**D**ynamic **R**andom **A**ccess **M**emory) един **бит се запаметява от кондензатор.** За достъп се използва един транзистор (1T1C-клетка).

DRAM технологията може да реализира повече памет на единица кристал от SRAM.

DRAM, обаче, е по-бавна от SRAM, защото данните в запаметяващите кондензатори трябва да бъдат периодично презаписвани (опреснявани).

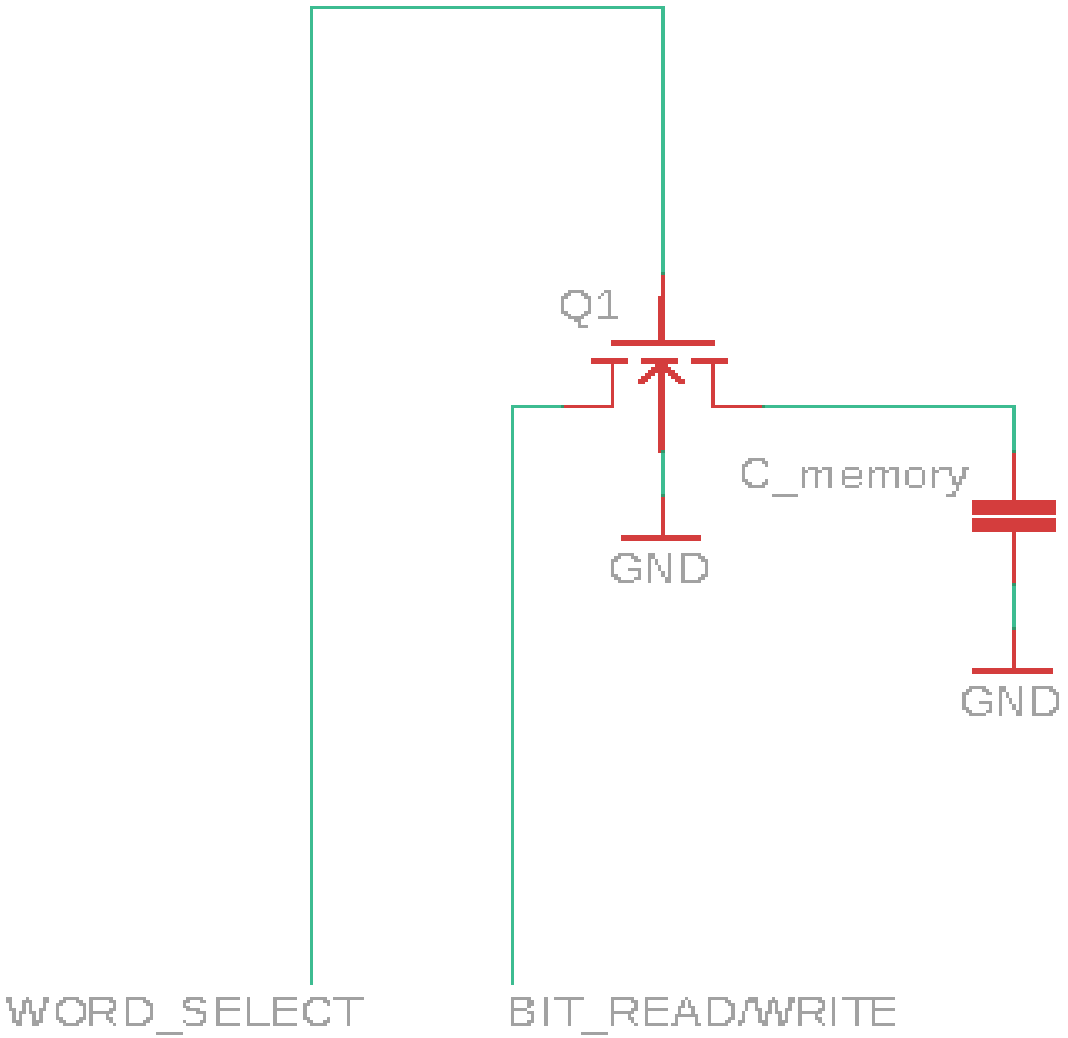
DRAM може да бъде достъпвана от μPU само през **контролер на паметта**.

DRAM е енергозависима памет.

Оперативна памет DRAM

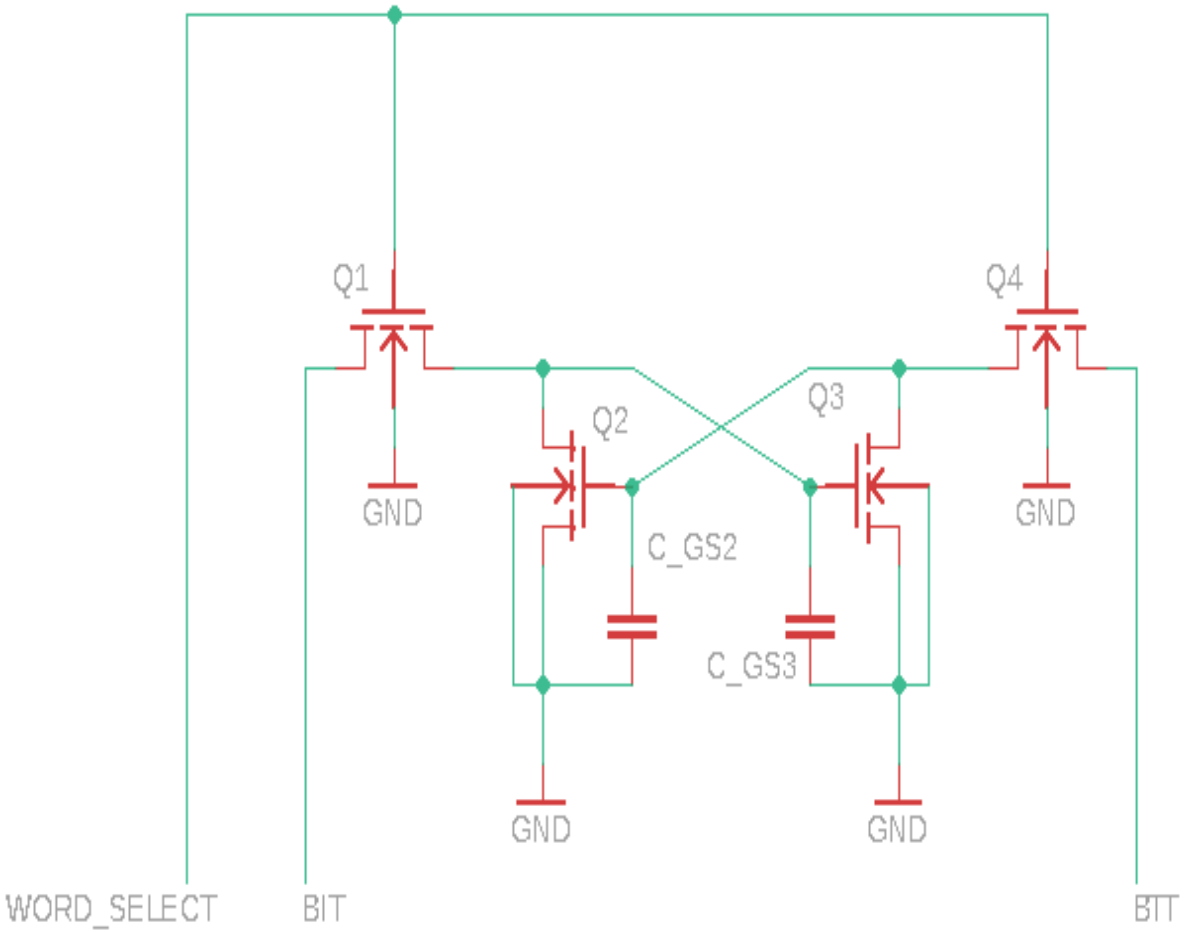
Исторически - първо се появяват DRAM клетки с 1 транзистор и 1 кондензатор (1T1C-клетка). Такава клетка е показана на схемата. Зареден кондензатор означава логическа 1, разреден – логическа нула.

Четенето е **разрушаващо**, т.е. трябва веднага след това да се презапише прочетения бит.



Оперативна памет DRAM

След това се появява 4T-клетката. При DRAM клетка с 4 транзистора (4T-клетка) се използват капацитетите C_GS на два NMOS транзистора.



Четенето е неразрушаващо, тоест:

1. При C_GS2 = 1, Q2 е буфер.
2. При C_GS3 = 1, Q3 е буфер.

Оперативна памет DRAM

За запис на 1:

==========

1. BIT = 1
2. WORD_SELECT=1
3. BIT1 линията зарежда C_GS3 на Q3 през Q1 (**C_GS3 = 1**)
4. Q3 и Q4 разреждат C_GS2 на Q2 (**C_GS2 = 0**).

За запис на 0:

==========

1. BIT = 0
2. WORD_SELECT=1
3. BIT1 линията разрежда C_GS3 на Q3 през Q1 (**C_GS3 = 0**)
4. Q4 зарежда C_GS2 на Q2 (**C_GS2 = 1**).

Оперативна памет DRAM

За четене на 1 (при начално условие **C_GS3 = 1, C_GS2 = 0**):

==========

0. Сигналите BIT и !BIT се зареждат до VDD
1. WORD_SELECT=1
2. !BIT ще се разреди до 0 през Q4 и Q3 (**!BIT = 0**)
3. Понеже Q1 и Q2 са запушени, BIT линията ще си остане заредена до 1 (**BIT = 1**)

За четене на 0 (при начално условие **C_GS3 = 0, C_GS2 = 1**):

==========

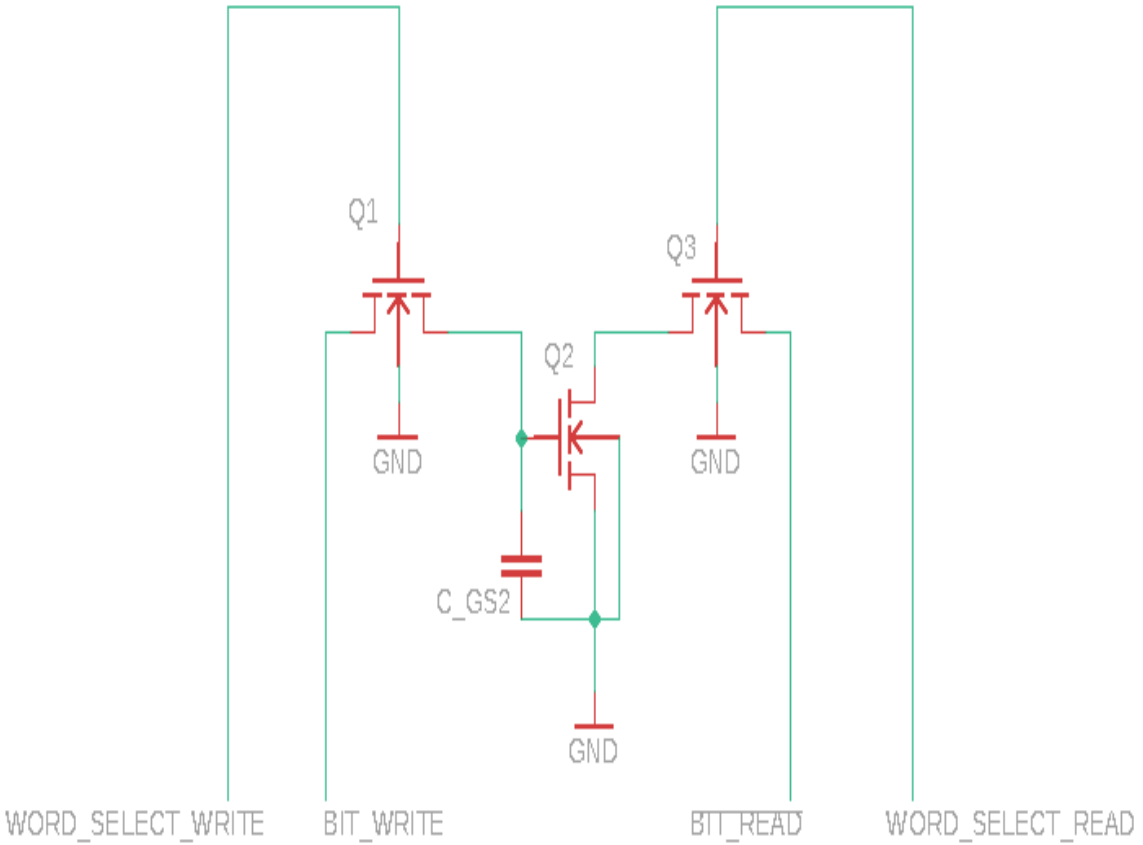
0. Сигналите BIT и !BIT се зареждат до VDD
1. WORD_SELECT=1
2. Понеже Q3 и Q4 са запушени, !BIT линията ще си остане заредена до 1 (**!BIT = 1**)
3. BIT ще се разреди до 0 през Q1 и Q2 (**BIT = 0**)

Оперативна памет DRAM

DRAM клетка с 3 транзистора (3T-клетка) е показана на схемата.

* Транзисторът Q1 извършва запис на 1/0, когато WORD_SELECT_WRITE = 1.
* Транзисторът Q3 извършва четене, когато WORD_SELECT_READ = 1.

Четенето е неразрушаващо, защото Q2 действа като буфер.



Оперативна памет DRAM

Стремежът е запомнящият кондензатор да **задържа заряд по-дълго**, което означава, че е необходим по-голям капацитет.

В същото време **размерите** на клетката трябва да са **възможно най-малки**.

Това не може да се постигне с планарна технология. Затова модерните DRAM клетки са само с **един специален транзистор** (1T-клетка), който включва в себе си и кондензатор. Това е т.нар. тренч-кондензатор (trench capacitor) [6].

Оперативна памет DRAM

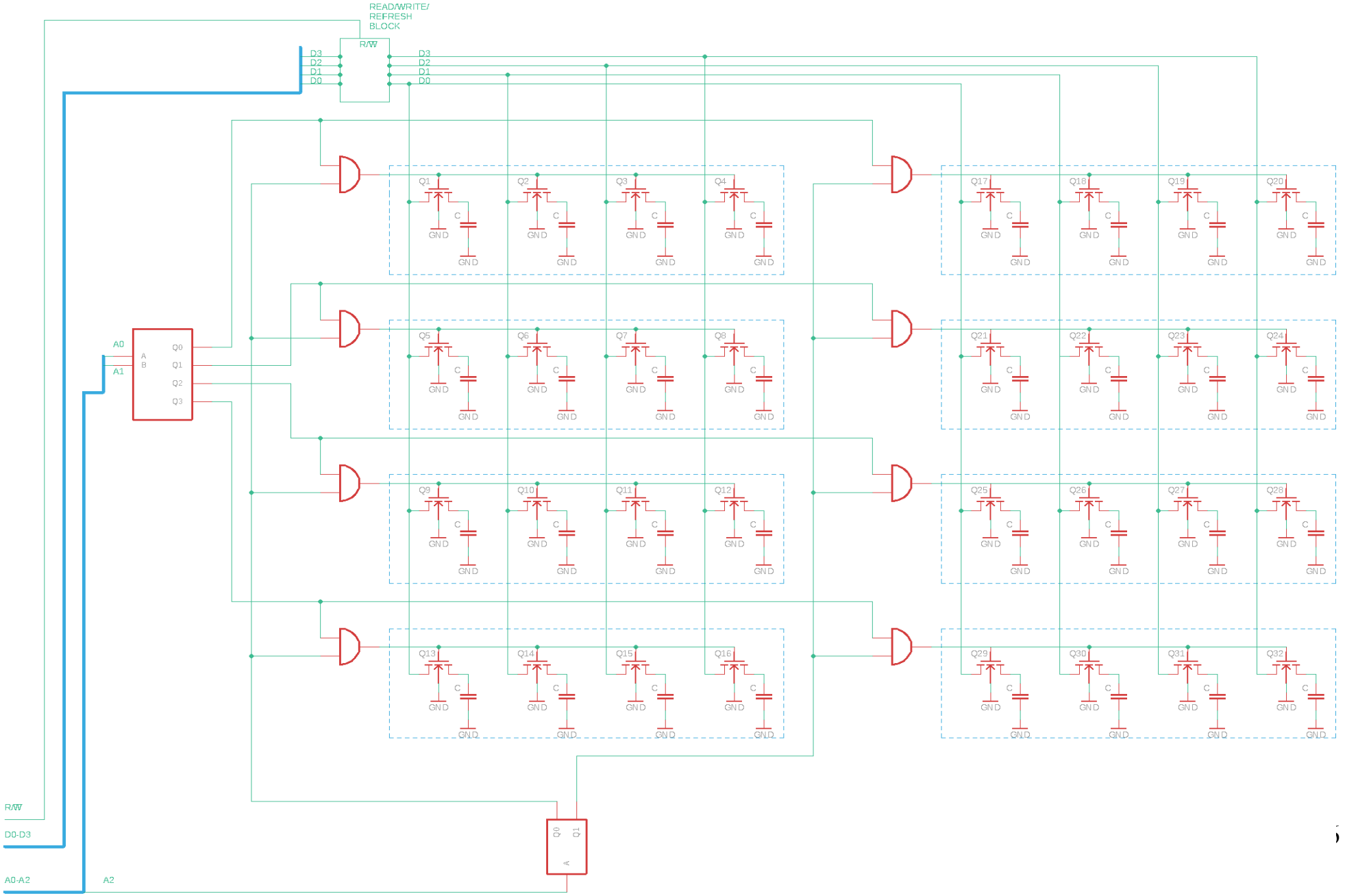
Капацитетът на битовата линия е много по-голям от капацитета на клетката. При четене, битовата линия се разрежда в незаредения кондензатор на клетката, но поради драстичните разлики в капацитета се получава много малко спадане на напрежението на битовата линия.

Затова са необходими **четящи усилватели/компаратори**. Различни литературни източници докладват **50 ÷ 250 mV** спадане на напрежението.

Необходима е допълнителна схемотехника за **презапис на бита след четене**.

Небходима е допълнителна схемотехника за **периодично опресняване** (**чрез четене**) на редовете и колоните от матрицата с битове.

Оперативна памет DRAM



Оперативна памет DRAM

Видове DRAM:

***DRAM** – запомнящите клетки са асинхронни (показаните дотук схеми)

***SDRAM** - запомнящите клетки се достъпват под управлението на един допълнителен сигнал – такт. Данните се достъпват само при единия фронт на такта.

***DDR SDRAM** – аналогична на SDRAM, но данните се достъпват и при двата фронта на такта.

Тази памет има няколко вида:

→ DDR-200, DDR-333, DDR-400 (100 – 200 MHz тактова честота)

→ DDR2-667, DDR2-800, DDR2-1066 (333 – 533 MHz тактова честота)

→ DDR3-1066, DDR-1333, DDR-1600 (533 – 800 MHz тактова честота)

→ DDR4-3200 (1600 MHz)

Оперативна памет DRAM

Видове DRAM:

***DRAM** – запомнящите клетки са асинхронни (показаните дотук схеми)

***SDRAM** - запомнящите клетки се достъпват под управлението на един допълнителен сигнал – такт. Данните се достъпват само при единия фронт на такта.

***DDR SDRAM** – аналогична на SDRAM, но данните се достъпват и при двата фронта на такта.

Тази памет има няколко вида:

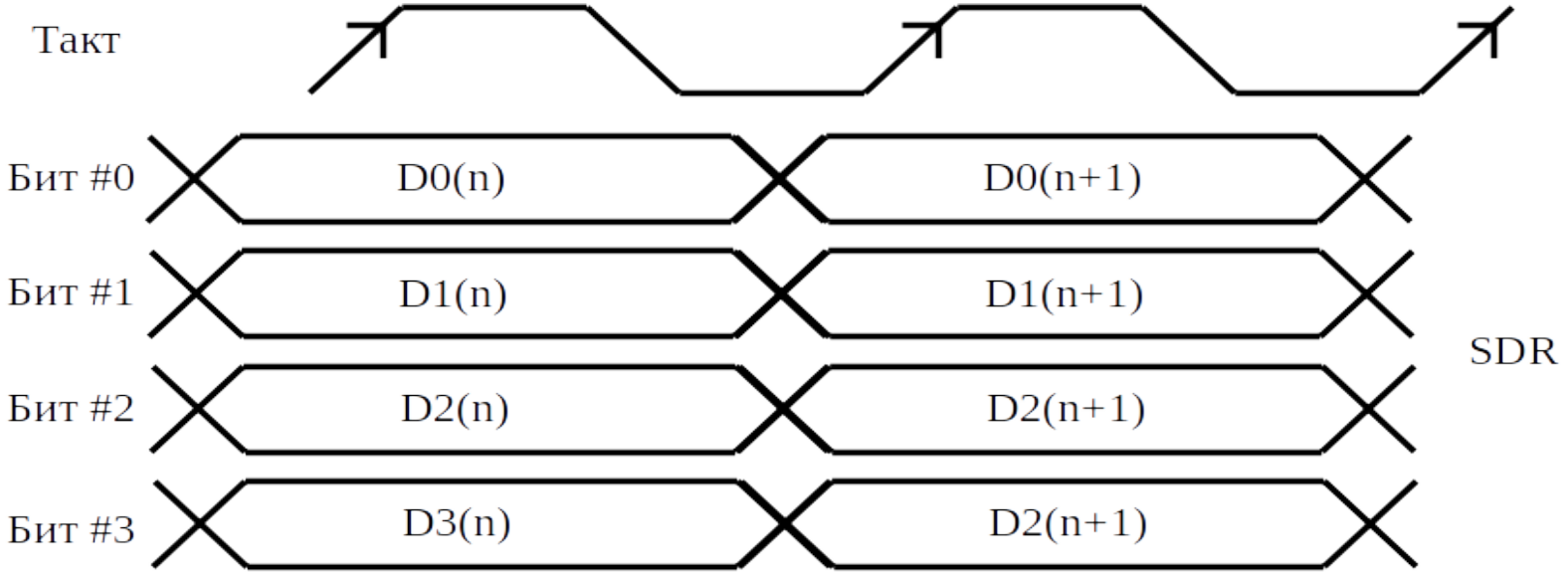
→ DDR-200, DDR-333, DDR-400 (100 – 200 MHz тактова честота)

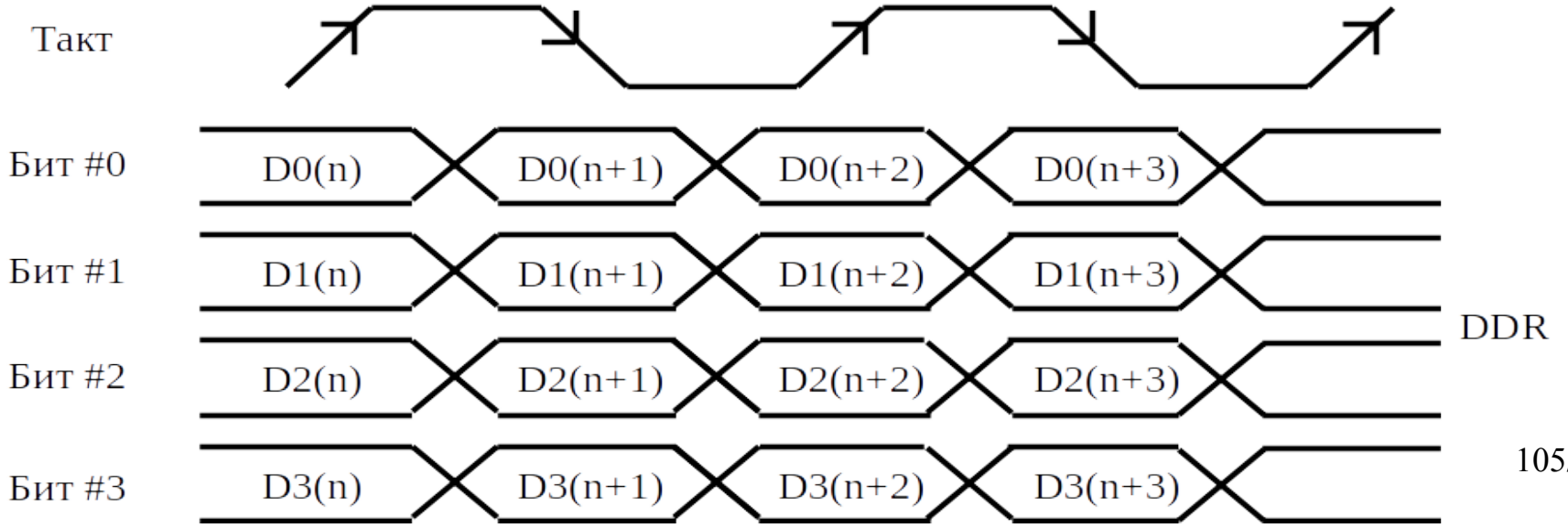
→ DDR2-667, DDR2-800, DDR2-1066 (333 – 533 MHz тактова честота)

→ DDR3-1066, DDR-1333, DDR-1600 (533 – 800 MHz тактова честота)

→ DDR4-3200 (1600 MHz)

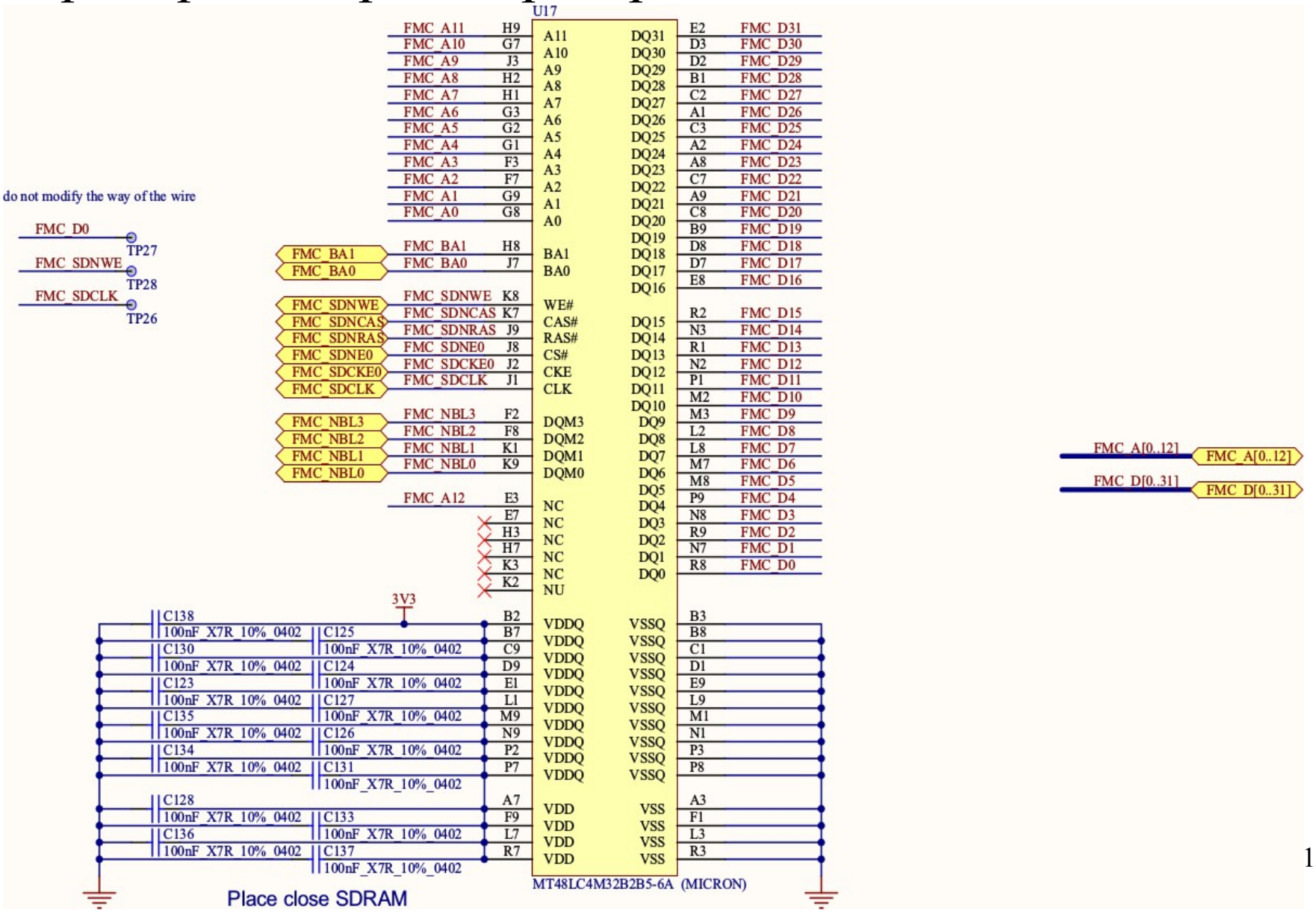
Оперативна памет DRAM





Оперативна памет DRAM

Пример – микроконтролер STM32F769 и SDRAM



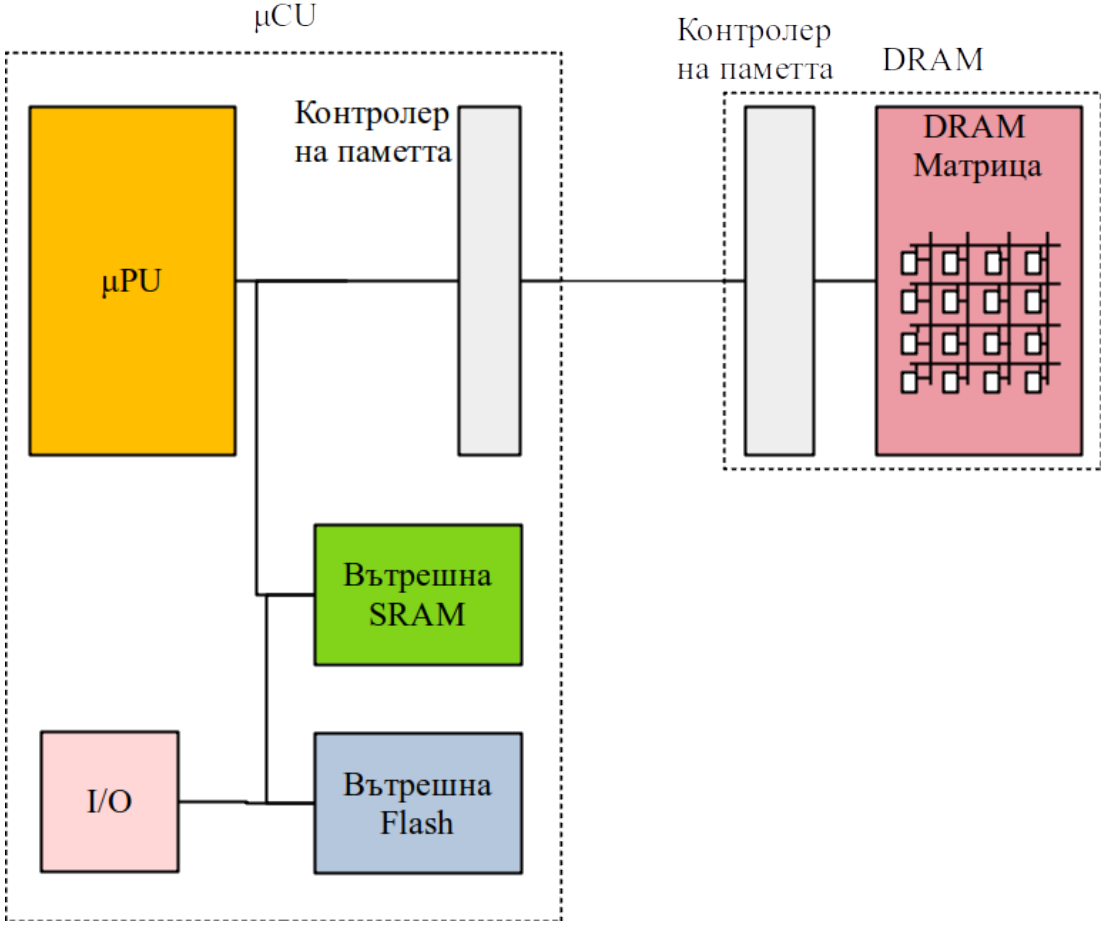
Оперативна памет DRAM

По-важни сигнали:

*CLK – тактов сигнал (нарастващ фронт)
*CS – избор достъп на данни/команди към чипа
*RAS, CAS, WE – участват при въвеждане на команди
*BA[1:0] – избор на банка (вътре в чипа)
*A[11:0] – 12-битови адреси
*DQ[31:0] – 32-битови данни
*VDDQ, VSSQ – захранване за данновите изводи
*VDD, VSS – 3.3-волтово захранване за вътрешната логика

Оперативна памет DRAM

Връзката между μPU и външна DRAM става посредством **контролери на паметта**. μPU "вижда" DRAM в адресното си поле, но всъщност между тях има два контролера на паметта – единият, от страна на μPU, задава команди, а другият, от страна на DRAM, изпълнява команди.



Оперативна памет DRAM

По-важни команди:

LOAD MODE REGISTER – конфигурира чипа при инициализация след подаване на захранването (дължина на burst-a, CAS времена, и др.)

ACTIVE – избери банка и ред (*не activate*)

PRECHARGE – зареждане на битовите линии и “изоставяне” на предишно-избрания ред

READ – избери банка и колона, и започни четене на много думи (burst)

WRITE - избери банка и колона, и започни запис на много думи (burst)

AUTO REFRESH – опреснява данните в кондензаторите. Тази команда трябва да бъде изпратена след PRECHARGE. За паметта от примера – трябва да се пуска 4096 пъти на всеки 64 ms.

Оперативна памет DRAM

Всички поддържани команди за конкретния чип. Забележете, че сигналите за адреси се използват за служебна информация в режим на команди.

Note 1 applies to all parameters and conditions

Name (Function)	CS#	RAS#	CAS#	WE#	DQM	ADDR	DQ
COMMAND INHIBIT (NOP)	H	X	X	X	X	X	X
NO OPERATION (NOP)	L	H	H	H	X	X	X
ACTIVE (select bank and activate row)	L	L	H	H	X	Bank/row	X
READ (select bank and column, and start READ burst)	L	H	L	H	L/H	Bank/col	X
WRITE (select bank and column, and start WRITE burst)	L	H	L	L	L/H	Bank/col	Valid
BURST TERMINATE	L	H	H	L	X	X	Active
PRECHARGE (Deactivate row in bank or banks)	L	L	H	L	X	Code	X
AUTO REFRESH or SELF REFRESH (enter self refresh mode)	L	L	L	H	X	X	X
LOAD MODE REGISTER	L	L	L	L	X	Op-code	X
Write enable/output enable	X	X	X	X	L	X	Active
Write inhibit/output High-Z	X	X	X	X	H	X	High-Z

Оперативна памет DRAM

По-важни параметри: tCAS-tRCD-tRP-tRAS

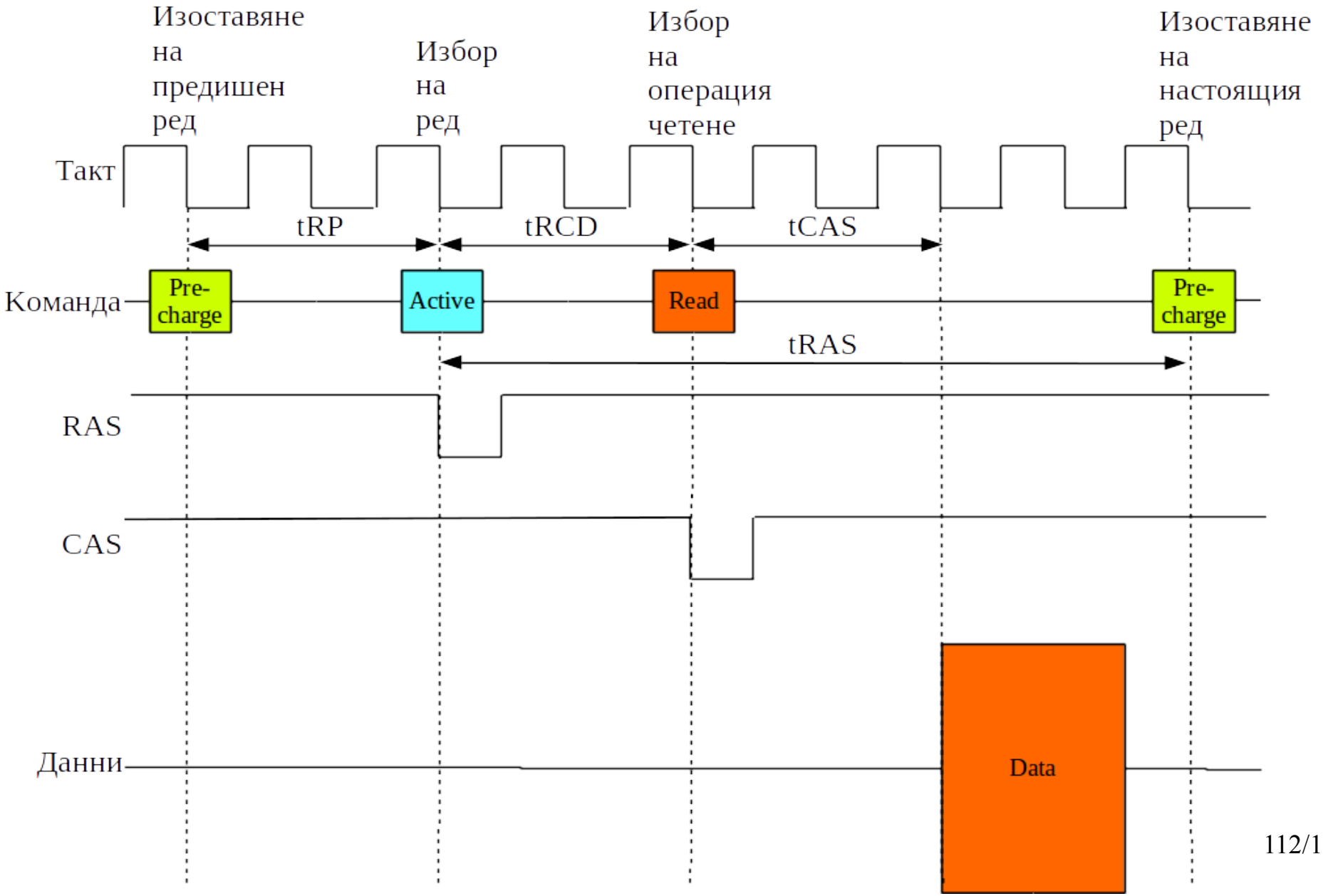
tCAS (Column Access Strobe, или още CL) – времезакъснението, в тактове, между команда READ и появяване на данните по данновата магистрала.

tRCD (RAS to CAS Delay) – минимално време, в тактове, между избор на ред и избор на колона.

tRP (RAS precharge) – времето, в тактове, между PRECHARGE и ACTIVE командите, т.е. времето между “изоставянето” на вече активен ред и “избирането” на нов.

tRAS (Row Access Strobe, Row Active Time) – минималното времезакъснение, в тактове, между ACTIVE и PRECHARGE команди.

Оперативна памет DRAM



Оперативна памет DRAM

Опресняването на паметта става ред по ред. Има няколко варианта *как да бъде стартиран този процес*:
*микропроцесорът сам издава команди за опресняване периодично (стар метод)
*контролер на паметта, вграден в микропроцесора
*брояч в самата памет

Съответните термини са:
*опресняване с RAS (RAS only refresh) – адресът на реда, който трябва да се опреснява се задава от контролера на паметта в микропроцесора
*опресняване с CAS преди RAS (CAS before RAS) – контролерът на паметта в микропроцесора издава команда за опресняване, а брояч в самата памет знае кой ред да опресни
*скрито опресняване (hidden refresh) – опресняване по време на четене/запис, докато данните се прехвърлят в изхода

Оперативна памет DRAM

Има два варианта *кога да бъде стартиран процеса* на опресняване:

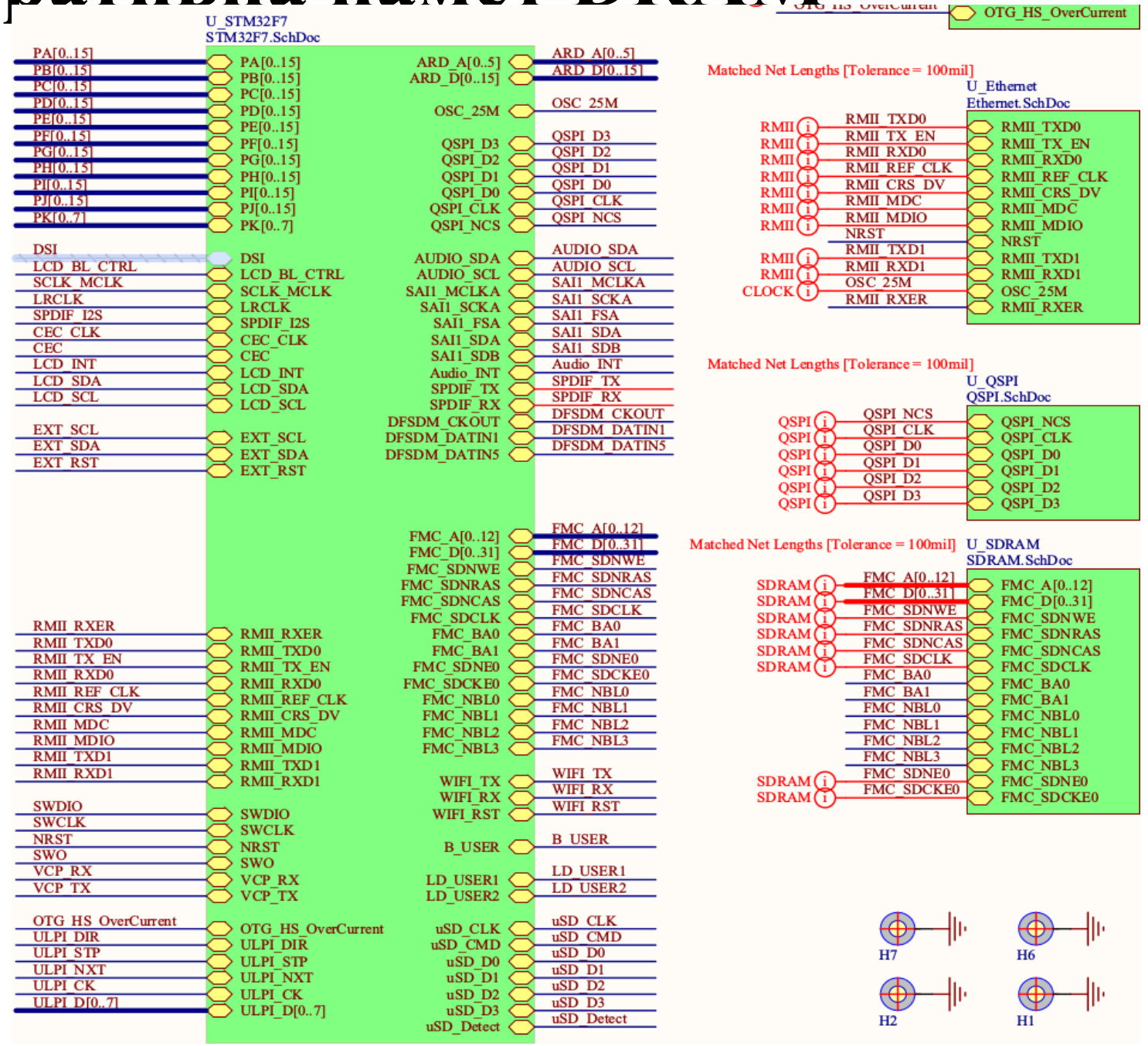
* цялостно (burst refresh) – всички редове се опресняват един след друг, докато не се стигне края на паметта. През това време микропроцесорът не може да я достъпва нито една клетка.

*разпределен (distributed) – опресняването на редовете става едновременно с достъпа до паметта. Микропроцесорът не може да достъпва клетки от реда, който в момента се опреснява.

Опресняването на клетките трябва да става веднъж на 4 ÷ 64 ms

Оперативна памет DRAM

Пистите на DRAM сигналите трябва да са с еднаква дължина, за да няма състезание на сигналите. Затова може да се видят “меандри” по платката в дизайни с DRAM.



Литература

[1] Г. Михов, “Цифрова схемотехника”, ТУ-София, 1999.
[2] K. Seong, “An Efficient High Voltage Level Shifter using Coupling Capacitor for a High Side Buck Converter”, http://dx.doi.org/10.5370/JEET.2016.11.1.125
[3] NXP Semiconductors, “Level shifting techniques in I2C-bus design”, application note AN10441, 2007.
[4] A. Islam, M. Hassan, “Variability Analysis of 6T and 7T SRAM Cell In Sub-45 NM Technology”, IIUM Engineering Journal, Vol. 12, No. 1, 2011.
[5] А. Керезов, “Ръководство за лабораторни упражнения по Микропроцесорна Схемотехника”, ТУ-София, 2000.
[6] А. Попов, “Импулсна схемотехника”, ТУ-София, 2016.