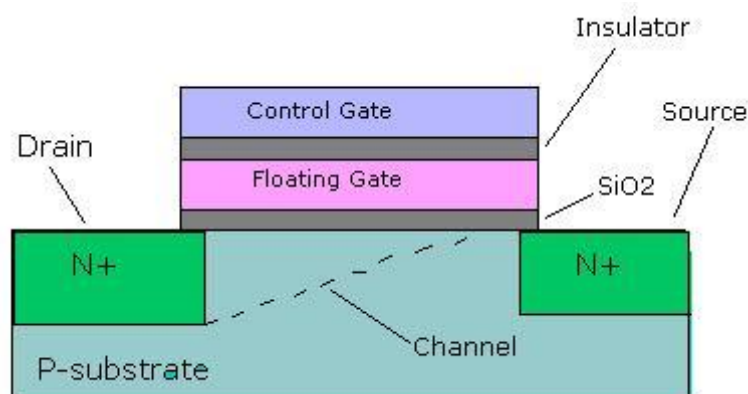


Технически Университет - Габрово	Организация на компютъра
Тема: Флаш памети	Лабораторно № 9
Цел: Запознаване с основните видове флаш (flash) памети, начина на функциониране и предназначението им при персоналните компютри	

## I. Теоретична част

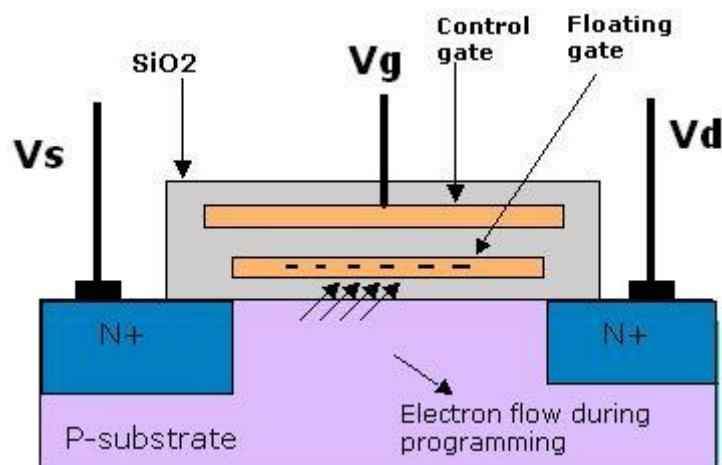
Флаш паметта е тип EEPROM (Electrically Erasable Programmable Read-Only Memory) – енергонезависима памет, която се изтрива и програмира по електрически път. Тази памет намира широко приложение в редица устройства: мемори карти, USB флаш памети, цифрови камери, мобилни телефони, цифрови плейъри, лаптопи и др. Флаш паметта е измислена от д-р Fujio Masuoka докато работи за Toshiba през 1980г. Според Toshiba, името "флаш" е предложено от колегата на д-р Masuoka, г-н Шоджи Ариууми, защото процесът на изтриване на съдържанието на паметта му напомня за светкавицата на фотоапарат. Д-р Masuoka представя изобретението си през 1984г. на IEEE International Electron Devices Meeting (IEDM), проведено в Сан Франциско, Калифорния. Първите NOR флаш памети са изобретени през 1984 година от сътрудници на фирма Toshiba. Toshiba анонсира NAND flash паметите на IEEE IEDM през 1987г.

Всяка запомняща клетка на флаш паметта е MOS транзистор с плаващ гейт (FGMOS). На фиг. 1 е показана структурата на FGMOS транзистор.

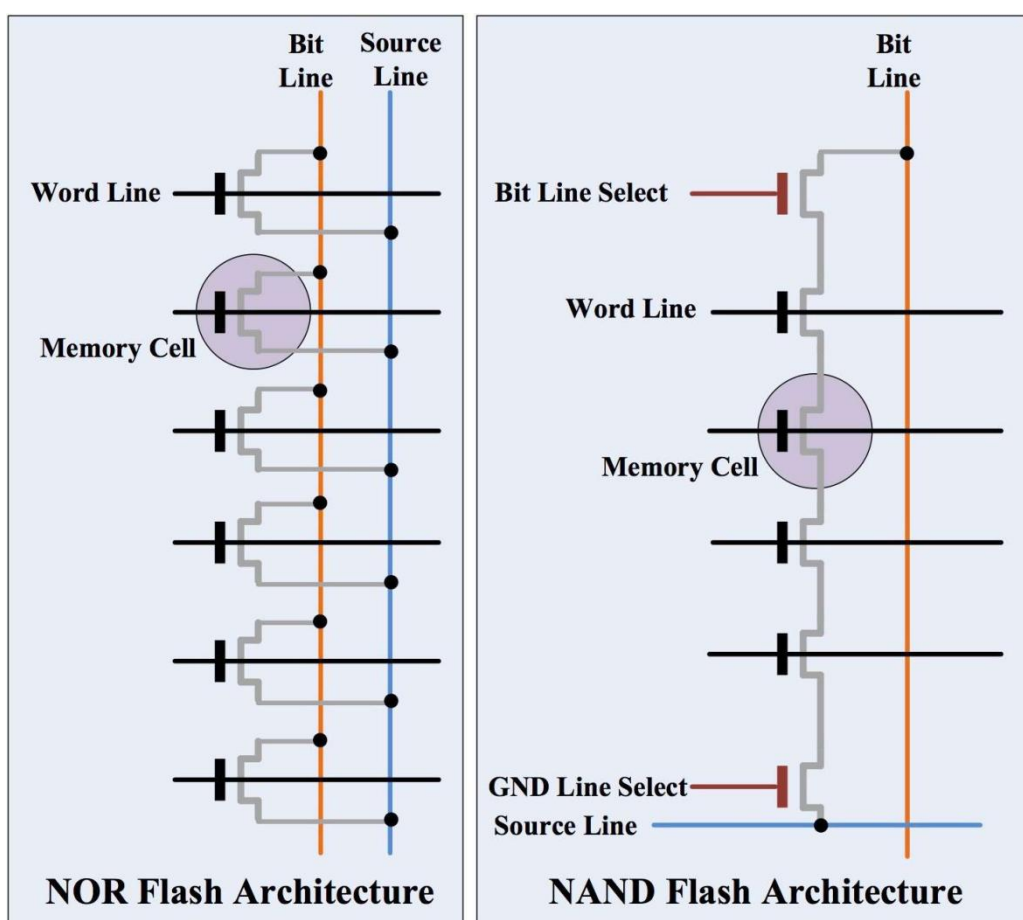


Фиг. 1. Транзистор с плаващ гейт

Транзисторът с плаващ гейт запомня един бит информация (логическа единица или логическа нула). Плаващият гейт е изграден в р-проводимата (наситена с дупки) област на управляващият гейт (този който управлява канала). Самият плаващ гейт е n-проводим (пропуска електрони) и е нанесен върху много тънък (около 10nm) слой диелектрик ( $\text{SiO}_2$ ). Този допълнителен гейт се използва за съхранение на електрони. Електроните в плаващият гейт могат да се съхраняват ограничен, но достатъчно дълъг период от време. Ако в него няма съхранени електрони, то при подаване на отпушващо напрежение към управляващият гейт, транзисторът ще се отпуши (това състояние съответства на лог. 0). Ако в него има съхранени електрони, то транзисторът ще остане запушен (това състояние съответства на лог. 1). Лесно може да се разбере дали има електрони в плаващият гейт или няма – подава се отпушващо напрежение на управляващият гейт и се проверява дали е протекъл ток между дрейна и сорта.



Фиг. 2. Натрупване на електрони в плаващия гейт



Фиг. 3. Структура на NAND и NOR флаш

Натрупването на заряд върху плаващия гейт (процес на запис) се реализира или чрез метода на инжекция на горещи електрони или чрез метода на тунелния ефект на Fowler-Nordheim. При метода на инжекция на горещи електрони, на сорса и на управляващия гейт се подава високо напрежение, което е необходимо за ускоряване на електроните до степен, при която те вече могат да преодолеят потенциалната бариера на тънкия изолационен слой (да го тунелират или “пробият”) и да достигнат областта на плаващия гейт. Ефективността на този процес обаче не е голяма и за постигане на желания ефект се налага да се приложи значително по-високо напрежение, така че между дрейна и сорса

да протече ток от порядъка на 1 mA (виж фиг. 2). За да се премахне натрупания в плаващия гейт заряд (процес на изтриване) на управляващия гейт се подава високо отрицателно напрежение, а на дрейна положително. С това се създава силно обратно електростатично поле и електроните се премахват от плаващия гейт, т.е. те тунелират обратно.

## **II. Видове флаш памети**

Двата основни типа флаш памет са NOR flash и NAND flash. Intel е първата компания, която въведе търговски NOR флаш чип през 1988г., а Toshiba пуска на пазара първата в света NAND флаш памет през 1989 г.

Имената NOR и NAND идват от структурата, използвана за междусистемните връзки между клетките на паметта (виж фиг. 3).

NOR флаш паметите са изградени от еднотранзисторни елементи подредени в матрица. Те образуват клетки и блокове. Запомнящите елементи се управляват по трите електрода – word линия, source линия и bit линия. Всички елементи от една клетка се управляват едновременно. Техните гейтове са свързани към общ проводник. Това означава че за достъп до даден елемент е нужно първо да се подаде положително напрежение на word линията на целия ред в който се намира. Информацията в дадения бит се прочита по съответната му bit линия – там са свързани сорсовете. Дрейновете са свързани към маса. NOR флаш паметите използват метода на инжекция на горещи електрони (който е описан по-горе). Схемата реализира логическата функция ИЛИ-НЕ (NOR). При прочит на един бит точно неговата стойност се появява на съответната bit линия, понеже останалите word линии са неактивни.

При NAND флаш паметите транзисторите са свързани последователно, което спестява много място. Сорсът е свързан към дрейнът на следващия транзистор в колоната. Гейтовете на всеки транзистор са свързани към word линията на дадения ред. За да се осъществи достъп до определен бит, първо се подава напрежение на съответната word линия – активират се всички транзистори от дадения ред. След това е нужно останалите транзистори да бъдат в такова състояние че да пропускат електричество по начин независим от техните стойности. Така при четене на данни безпроблемно протича електричество по цялата серия последователно свързани транзистори, до достигане на избрания бит за четене. По този начин информацията от избраните битове за четене се появява на bit линията. Предимства на тази архитектура са бързият запис и изтриване на данни и по-добрата мащабируемост.

## **III. Предимства и недостатъци на NOR и NAND флаш**

Предимства на NOR flash паметите са бързия достъп и прочитане на дадена клетка. Недостатъци са бавният запис и изтриване. Например, времето за изтриване на блок NOR флаш с размер 128KB блоке около 750ms, докато при NAND памет със същия капацитет това време е около 2ms. NAND-flash има няколко пъти по-бързо изтриване и записване при сравнение с NOR флаш. Тъй като NAND-flash се изтрива на блокове са необходими по-малко цикли на изтриване при сравнение с NOR-flash. NOR-flash чете данните малко по-бързо от NAND-flash.

NOR паметите имат адресна и шина за данни които позволяват произволен достъп до всяка клетка (байт) на паметта. Това прави тази памет подходяща за замяна на по-стари чипове на BIOS, които рядко трябва да се актуализират. Броят на циклите на изтриване е от 10,000 до 1,000,000. NOR паметта е подходяща за съхранение на програмен код при вградени системи. В този случай програмният код може да се стартира директно от NOR паметта.

NAND паметта заема по-малка площ на чипа при сравнение с NOR паметта. NAND-flash позволява по-голяма плътност на съхранение на данните при по-малки производствени разходи за бит информация при сравнение с NOR-флаш. Освен това има до десет пъти по-голяма издръжливост на NOR флаш. NAND е по-подходящ като носител за съхранение на големи файлове, включително видео и аудио. USB флаш, SD и MMC картите са от тип NAND флаш. Основен недостатък на NOR флаш е лошата мащабируемост. Тя се дължи на това че към всеки запомнящ елемент е нужно да се подаде отделен контакт.

NAND-flash паметите нямат външна адресна шина с произволен достъп, така че данните трябва да се четат на блокове (страници), където всеки блок съдържа стотици хиляди бита. Това е една от основните причини, поради която NAND-flah не е подходяща за замяна на ROM памети, тъй като повечето микропроцесори и микроконтролери изискват байт-ориентиран достъп до паметта.

#### IV. Задачи за изпълнение

##### Задача 1.

Направете сравнителен анализ на NOR и NAND flash памети. Анализирайте следните характеристики:

- Максимален капацитет;
- Време за изтриване;
- Време за програмиране;
- Време на достъп (четене);
- Цена при един и същи капацитет;
- Размери на чипа при един и същи капацитет;
- Възможност за директно изпълнение на програмен код;
- Консумация;
- Брой презаписи.

Систематизирайте информацията в таблица, както е показано на фиг. 4.

Parameter	NOR	NAND
Density	1 Mbit – 1 Gbit	64 Mbit - 1 Gbit
Read initial access	55 ns	10,000 ns
Read sequential access	9 ns	50 ns
Program	0.3 Mbytes/s	2.6 Mbytes/s
Erase	0.2 Mbytes/s	8.2 Mbytes/s
Access Method	Random	Sequential

Фиг. 4. Сравнителен анализ на NOR и NAND флаш памети

## **Задача 2.**

Каква е разликата между SLC и MLC флаш памети?

## **Задача 3.**

Направете сравнителен анализ между различните видове флаш-карти памет (CompactFlash, MMC, SmartMedia, MemoryStick, xD, SecureDigital) като ги сравните по следните параметри:

- максимална скорост на обмен при четене и запис в MB/s;
- тип на паметта (NOR или NAND);
- област на приложение.

## **Задача 4.**

Какви основни компоненти изграждат един USB flash drive? Какво е тяхното предназначение? Какъв тип флаш памет се използва – NOR или NAND?

## **Задача 5.**

NAND паметите са достъпни на ниво блок. Всеки блок се състои от страници, всяка от които е с определен размер, например 512B или 2KB. За всяка страница се заделени по 12 или 16 байта за ECC. Какво е тяхното предназначение? Колко бита от една страница могат да бъдат коригирани чрез ECC?

## **Задача 6.**

Какви файлови системи за флаш памети съществуват?

## **Задача 7.**

Какво представляват SSD носителите? Какъв тип флаш памет се използва при тях и защо?

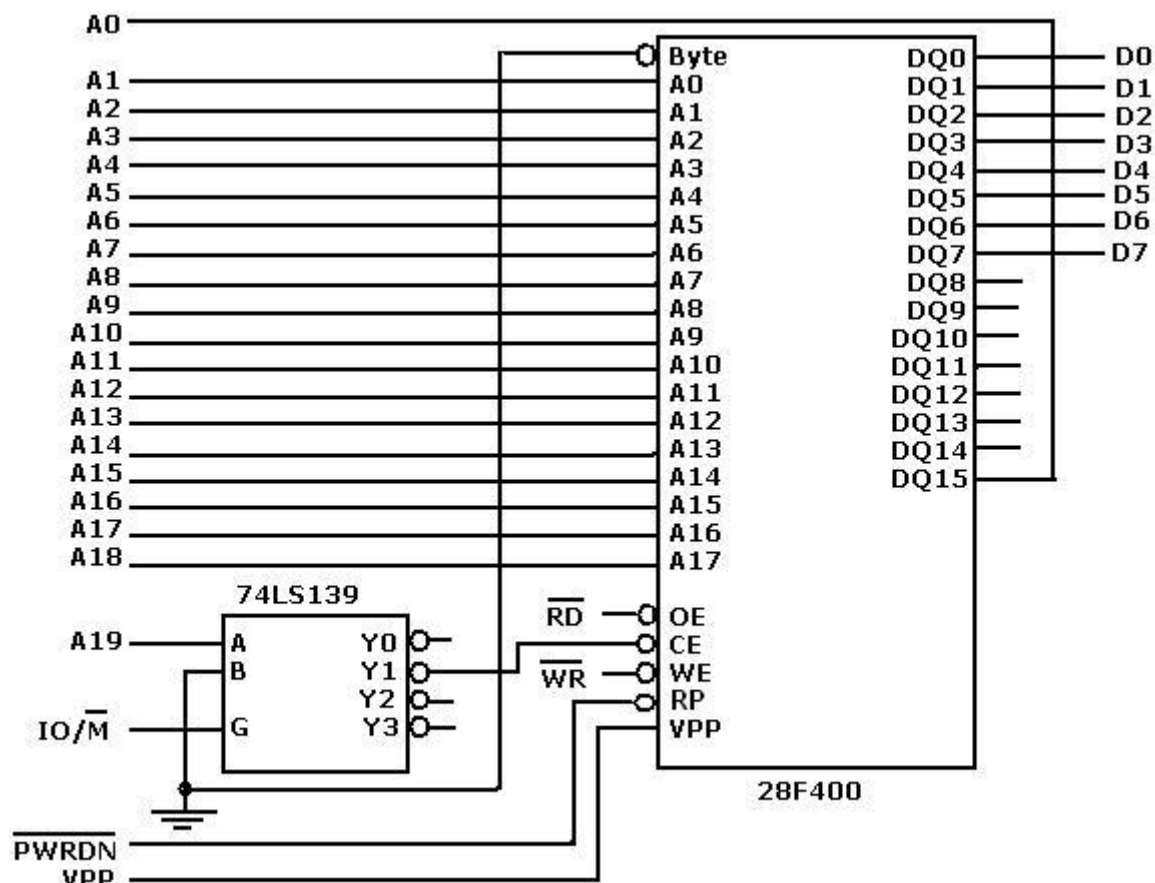
## **Задача 8.**

NAND клетките се записват чрез процес наречен тунелно инжектиране, а се изтриват чрез тунелно освобождаване. Какво представляват тези процеси?

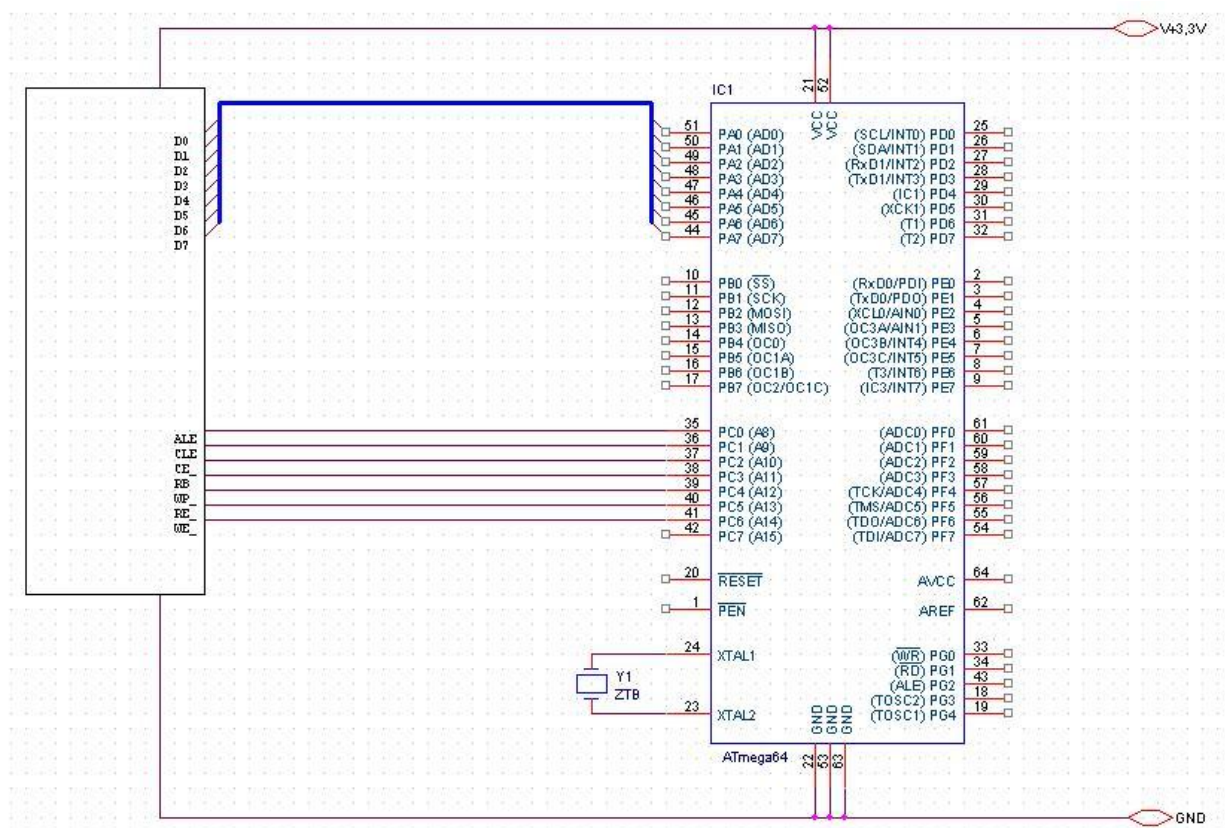
## **V. Допълнителни въпроси**

1. На фиг. 5 е показано примерно управление на NOR флаш памет 28F400 на фирма Intel. Тази памет може да бъде конфигуриран като 512K x 8 памет или като 256K x 16 памет. Как е конфигурирана паметта при конкретното свързване? Какво е предназначението на управляващите сигнали OE, CE, RP и Byte?

2. На фиг. 6 е показано примерно управление на NAND флаш памет K9K8G08U0A на фирма Samsung. Тази памет позволява изтриване само на блок данни или на целия чип; запис на цяла страница и четене на цяла страница. Какъв е размерът на паметта? Ако трябва да промените съдържанието на един байт от страница, как трябва да се процедира? Какво е предназначението на управляващите сигнали ALE, CLE, CE, RE, R/B, WP, RE и WE? Опишете реализацията на команди Write и Erase.



Фиг. 5. Управление на NOR флаш памет



Фиг. 6. Управление на NAND флаш памет