



Универзитет „Св.Кирил и Методиј“ – Скопје  
Факултет за електротехника и информациски  
технологии



Проектна задача по предметот

Микропроцесорка електроника

# Скенер на матрична тастатура

**Изработил**

Стефан Василевски, 276/2012

**Предметен професор**

Проф. д-р Зоран Ивановски

Скопје, 2024



## Скенер на матрична тастатура

Во оваа проектна задача се изработува скенер на матрица од копчиња составена од 3 колони и 4 редови кој симулира внесување на лозинка во меморија.

Во програмата користиме бројач(претходно зададен од програмерот-но и тој може да се ре-програмира) кој дозволува скенирања од матрицата(тастатурата) и зачувување во меморија, подоцна внесената лозинка ја споредуваме со наша претходно зачувана лозинка во меморијата.

Зачувувањето во меморија од исчитувањето и зададената лозинка за споредба се во две ASCII низи од карактери.

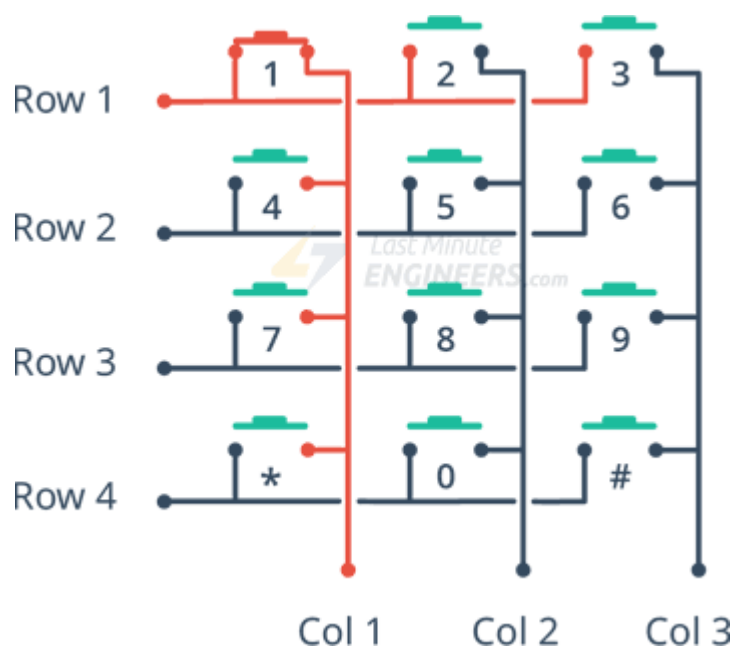
Матричната тастатура е составена од мрежа од прекинувачи(тастери). За 3x4 тастатура, има 3 колони и 4 редици, вкупно 12 прекинувачи. Секој прекинувач е наместен на пресекот на специфична редица и колона.

Постојат седум спроводника од кои три за колоните и четири за редиците.

Секој прекинувач(тастер) спојува специфична колона со специфичен ред.

При притиснување на прекинувач, формирана е конекција помеѓу специфичен ред и специфична колона.

За пример, доколку го притиснеме копчето со реден број '1', колото се затвора помеѓу редот број 1 и колоната број 1.(види слика доле)



Внатрешна шема на поврзаните тастери и пример

Кодот е составен од повеќе делови и има една функција. Деловите од кодот се:

Поставување на комбинацијата на излезот од паралелната порта.

Читање(Скенирање) на влезот од паралелната порта и споредување на точната комбинација(декодирање).

Доколку е притиснат тастер, во зависност од споредената комбинација добиена на влез се запишува во меморија преку покажувач соодветен ASCII код и истовремено испишување на екран “скенираното” копче.

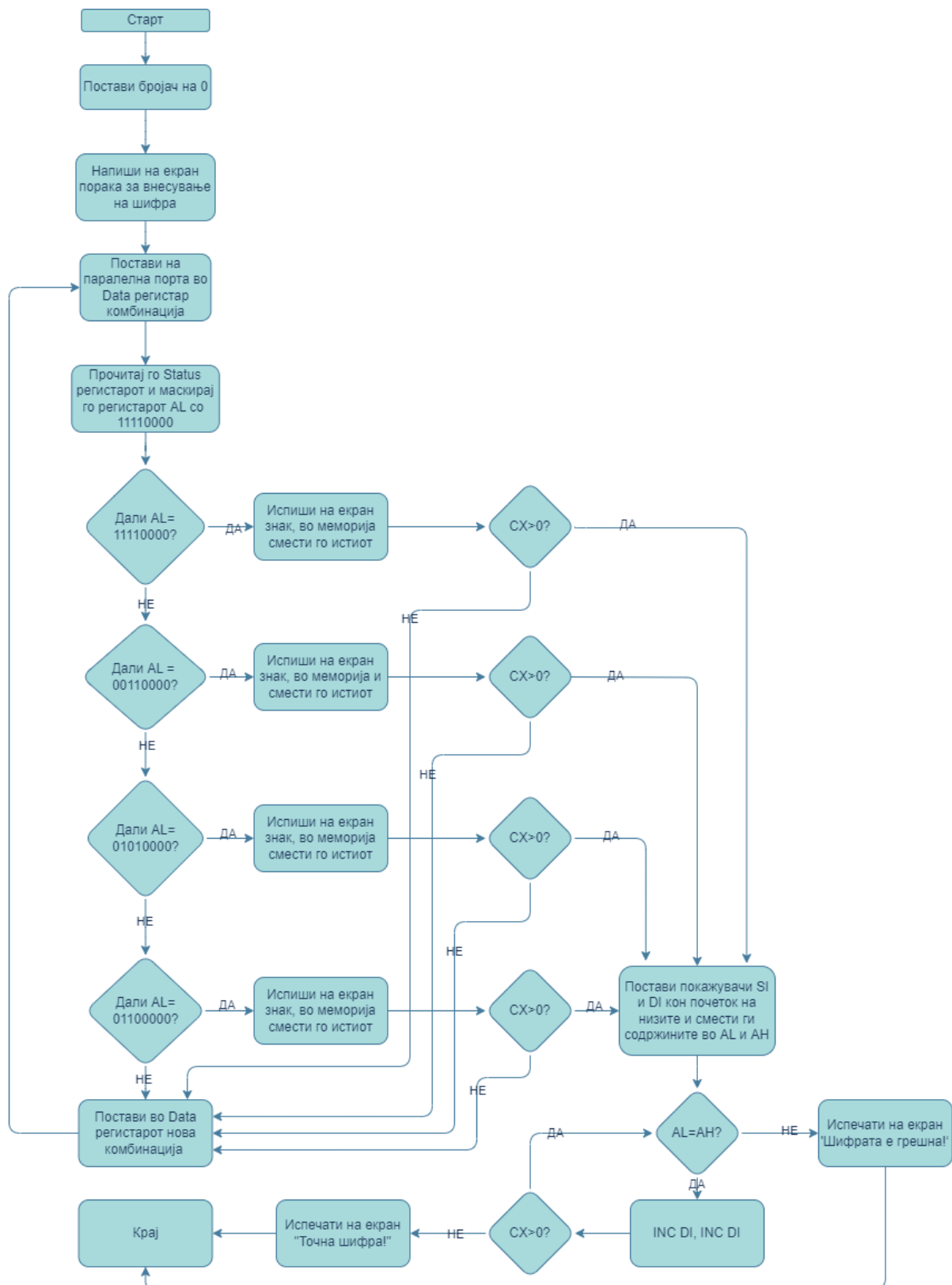
Споредба на низата од карактери(низа шифра и низа внесена шифра)

Функција Debounce delay

		Data register		
		D2	D1	D0
Status register	Select	"1"	"2"	"3"
	Paper	"4"	"5"	"6"
	Ack	"7"	"8"	"9"
	Busy	"*"	"0"	"#"

Слика на поврзување на матричната тастатура со LPT1 портот

Пред да ги објасниме овие делови од кодот подетално, да го нацртаме блок дијаграмот на скенерот:



Алгоритам на програмата за исчитување на матричната тастатура

*Поставување на соодветна комбинација на паралелна порта* во излезниот Data регистар. Битови кои се користат се пониските три (LSB) од D0-D2 соодветно поврзани на трите хоризонтални линии на матричната тастатура (колони).

### **Читање на Status регистарот**

Читањето на Status регистарот е со цел да ја добиеме соодветната комбинација на некој притиснат тастер. Исчитувањето од паралелната порта се врши со IN инструкција, а подоцна се врши маскирање на AL регистарот. Доколку корисникот притисне два или повеќе тастери на тастатурата во ист момент, самата CMP инструкција ќе го “отфрли” и ќе продолжи кон друга лабела која споредува дали комбинацијата за некој друг тастер е добиена од прочитаниот регистар, во спротивно тастерот е притиснат и покажувач кој покажува во меморијата запишува ASCII код. Во кодот се наоѓа LOOP инструкција која дозволува запишување во меморија и на екран толку притисоци на тастери колку што е CX.

### **Споредба на низата од карактери**

Споредбата на низата од претходно внесените карактери со некоја зададена низа од карактери (цифри и знаци – ‘#’ и ‘\*’ , тараба и ѕвезда) се врши со два покажувачи, претходно иницијализирани да покажуваат кон почетокот на двете низи од карактери кои се наоѓаат во меморијата. Со CX и LOOP (и DEC CX со JNZ поради тоа што LOOP не може да направи поголем скок) инструкцијата се изминува и врши споредба на внесената со некоја точна шифра претходно внесена од корисникот. Во зависност дали секој карактер од низата и нивна споредба, програмата одлучува дали е точна или не точна шифрата. Ако LOOP инструкцијата заврши со одбројување на CX, тогаш низата е точна, доколку не е точна одлучува JNE инструкцијата која претходно ги споредува карактерите кои се поклопуваат во двете низи.

### **Функција Debounce delay (Процедура)**

Debounce delay е функција која го запира читањето (од Status регистарот) за одредено време. Доцнењето функционира со регистарот ECX и DEC инструкција за проверка дали регистарот ECX е испразнет. CX претходно е искористен, но поради тоа што времето постигнато со CX е многу мало функцијата Debounce delay не е ефективна.