

İ.T.Ü.
Bilgisayar-Bilişim Fakültesi
Bilgisayar Mühendisliği Bölümü



MİKROBİLGİSAYAR
LABORATUVARI
DENEY RAPORU

Deney No : CSM12C32_Deney2
Deney Adı : Kesme Uygulamaları
Deney Tarihi : 01.12.2010
Grup : 10
Deneyi Yapanlar : 040080153 Serkan Güler
040080322 Osman Boyacı
040090533 Abdullah Aydeğer

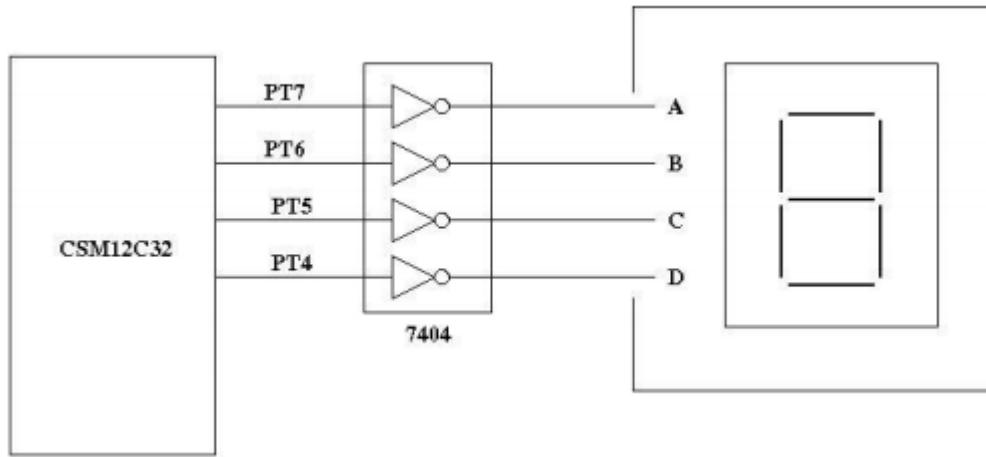
Deneyi Yaptıran Araştırma Görevlisi : Hasan Kıvrak

Deneyin Amacı:

Bu deneyde, kesme kavramının öğrenilmesi, kesme, kesme isteği ve yazılım kesmesi kavramlarının anlaşılması, kesme hizmet programı geliştirme ve titreşimsiz düğme kavramının irdelenmesi ve geliştirilmesi amaçlanmıştır.

BÖLÜM 1:

Yazılması istenen programda bir yön değişkeninin değerine göre program ya ileri doğru ya da geri doğru sayacaktır. Program ileri doğru 9'a kadar, geri 0'a kadar sayacaktır.



Bu donanım üzerinde çalışması için yazdığımız program;

```
XDEF Entry
ABSENTRY Entry
INCLUDE 'mc9s12c32.inc'
ORG $3800
Counter1    DC.W  $000F
Counter2    DC.W  $0010
Entry       LDAA   #$F0
            STAA   DDRT
DirectionInit LDAA  #$01
            STAA   $3C00
DirectCheck LDAA   $3C00
            CMPA   #$00
            BEQ    Backward
```

Forward LDAA #\$00
FwdLoop COMA
 STAA PTT
 COMA
 ADDA #\$10
 JSR Delay
 CMPA #\$A0
 BEQ Forward
 BRA FwdLoop

Backward LDAA #\$90
BwdLoop COMA
 STAA PTT
 COMA
 SUBA #\$10
 JSR Delay
 CMPA #\$F0
 BEQ Backward
 BRA BwdLoop

Delay LDY Counter2
DelayLoop JSR SubDelay
 DEY
 BNE DelayLoop
 RTS
SubDelay LDX Counter1
SubDlyLoop NOP
 DEX
 BNE SubDlyLoop
 RTS

BÖLÜM 2:

Kesme isteği geldiğinde yön değişkenini diğer yönde sayma yapacak şekilde değiştirecek programın kodu aşağıda verilmiştir.

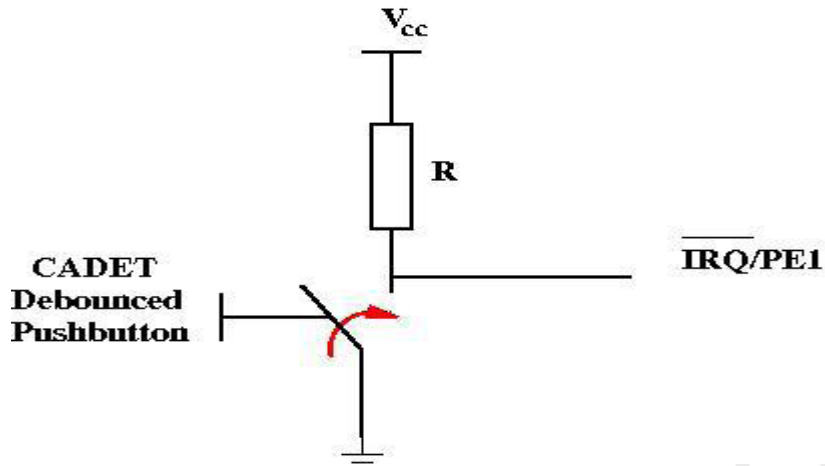
```
XDEF Entry
ABSENTRY Entry
INCLUDE 'mc9s12c32.inc'
ORG $3800

Counter1    DC.W  $000F
Counter2    DC.W  $0010
Entry        LDS   #StackST
             LDAA  #$F0
             STAA  DDRT
             SEI
             BSET  INTCR, mINTRC_IRQE
             BSET  INTCR, mINTRC_IRQEN
             CLI
DirectionInit LDAA  #$01
             STAA  $3C00
DirectCheck LDAA  $3C00
             CMPA  #$00
             BEQ   Backward

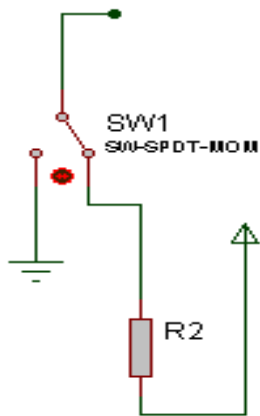
Forward      LDAA  #$00
FwdLoop      COMA
             STAA  PTT
             COMA
             ADDA  #$10
             JSR   Delay
             CMPA  #$A0
             BEQ   Forward
             BRA   FwdLoop
```

Backward	LDAA #\$90
BwdLoop	COMA
	STAA PTT
	COMA
	SUBA #\$10
	JSR Delay
	CMPA #\$F0
	BEQ Backward
	BRA BackwardLoop
Delay	LDY Counter2
DelayLoop	JSR SubDelay
	DEY
	BNE DelayLoop
	RTS
SubDelay	LDX Counter1
SubDlyLoop	NOP
	DEX
	BNE SubDelayLoop
	RTS
ISR1	LDAB \$3C00
	CMPB #\$00
	BEQ ChDirFwd
ChDirBwd	LDAB #\$00
	STAB \$3C00
	RTI
ChDirFwd	LDAB #\$01
	STAB \$3C00
	RTI
	ORG \$FFF2
	DC.W ISR1

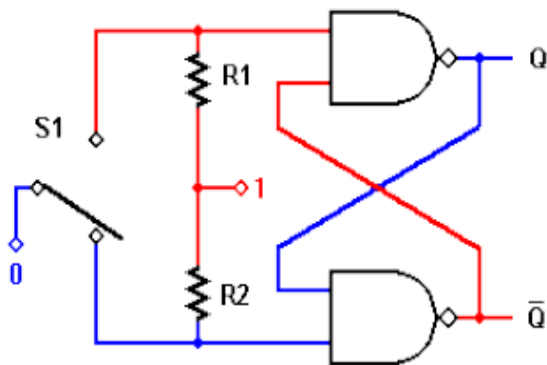
► Gerçeklenilen kesme donanımı:



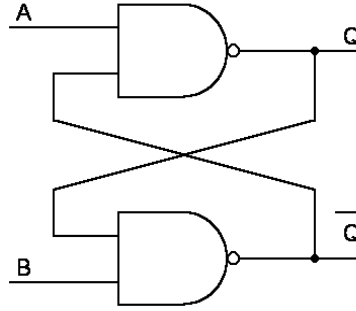
(SPDT anahtarla) İkinci kesme devresi:



Titreşimsiz düğme tasarımı (üçüncü kesme devresi):



► Titreşimsiz anahtar devresinin açıklaması:



S'(A)	R'(B)	Q	Q'
0	1	1	0
1	1	1	0
1	0	0	1
1	1	0	1
0	0	1	1

TVE kapıları ile gerçekleştirilen tutucunun doğruluk tablosuna bakılınca iki girişe de lojik 1 verildiği zaman önceki durumu koruduğu görülüyor. İki girişe de lojik 0 verilmesi durumu yasaklı durumdur. Düğmeye basıldığında titreşimler iki girişe de lojik 1 değerini verse bile bu girişlerin önceki durumu koruduğu ve çıkışı etkilemediği bilindiğinden anahtar yön değiştirirken dalgalanma olduğunda ilk defa konum değişir değişmez bu konumun tutulacağı görülür.

► Yazılım kesmesi kavramı:

Yazılım kesmesi adı üzerinde kesmenin dışardan değil de yazılımla (kesme komutu) yapılması durumudur. Yazılım kesmesi bir altprogram gibidir ama kullanılış biçimi altprogramdan farklıdır. Altprogramda, altprogramın çağrıldığı adres yığına atılırken yazılım kesmesinde, mikroişlemci içindeki kütüklerin değeri de yığına atılır, buradan çıkışta tekrar eski değerler yığından çekilir. Bu yüzden kesme programlarında kütükler üzerinde yapılan işlem saklanmak istenirse, bunların bellekte belli bir adrese yazılması ve gerektiği zaman tekrar bu bellekten okunması gerekmektedir.

```

XDEF Entry
ABSENTRY Entry
INCLUDE 'mc9s12c32.inc'
ORG $3800

Counter1    DC.W $000F
Counter2    DC.W $0010
Entry       LDS  #StackST
            LDAA  #$F0
            STAA  DDRT
            SEI
            BSET  INTCR, mINTRC_IRQE
            BSET  INTCR, mINTRC_IRQEN
            CLI
Init        LDAA  #$00
            STAA  $3D00

InfiniteLoop LDAA  #$3D00
            SEI
            COMA
            STAA PTT
            COMA
            CLI
            JSR  Delay
            BRA  InfiniteLoop

ISR1        LDAA  $3D00
            COMA
            CMPA  #$90
            BEQ  Reset

Increment   ADDA  #$10
            COMA
            BRA  InfiniteLoop

```


Reset LDAA #\$00
 COMA
 BRA InfiniteLoop

Delay LDY Counter2

DelayLoop JSR SubDelay
 DEY
 BNE DelayLoop
 RTS

SubDelay LDX Counter1

SubDlyLoop DEX
 BNE SubDlyLoop
 RTS

ORG \$3FFF2

DC.W ISR1