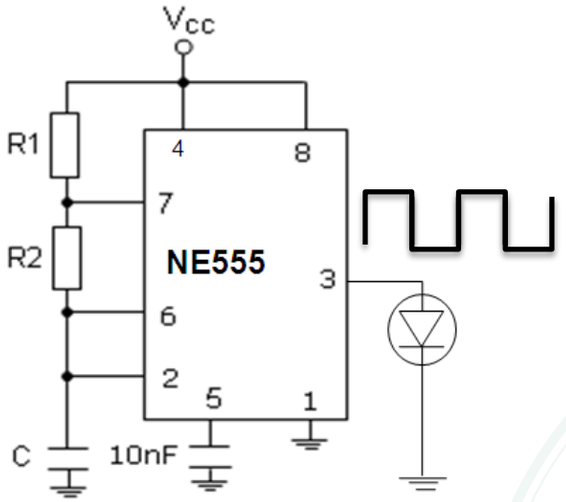
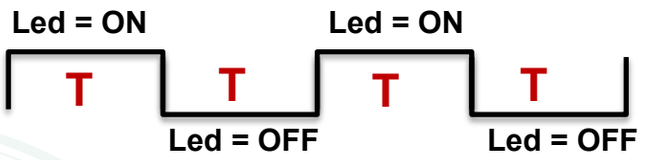
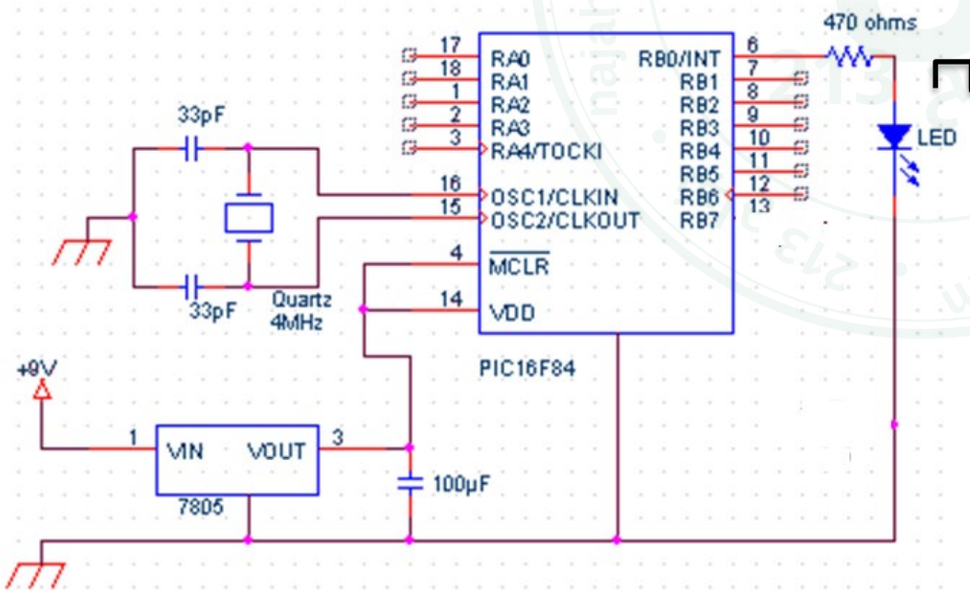
التأجيل بواسطة الحلقات

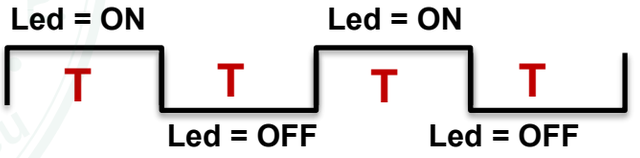
فروض واختبارات مع الحلول لكل المستويات — مجاناً

تذكير

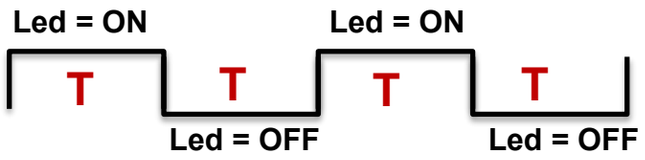


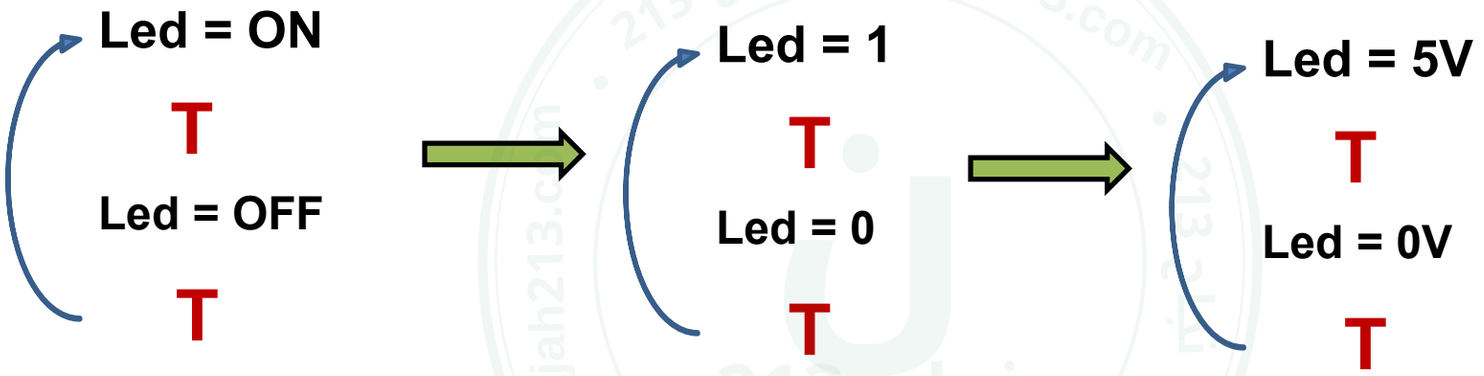


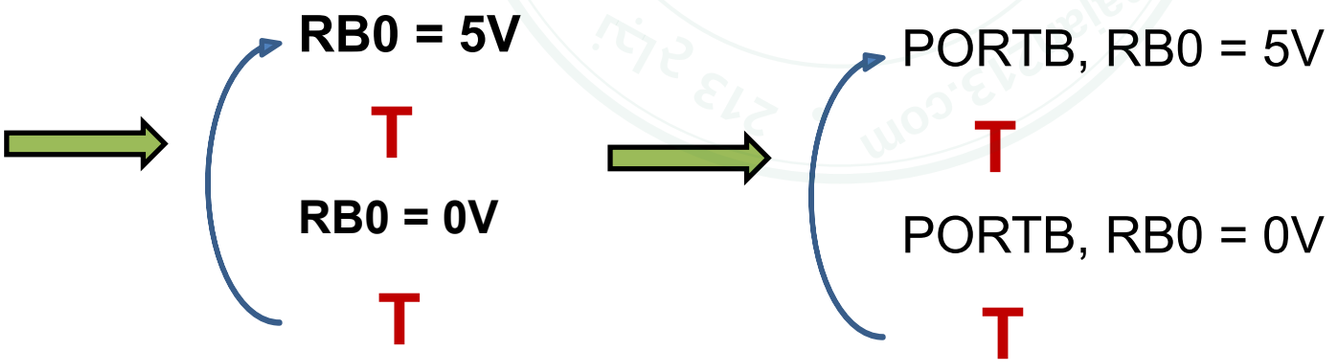




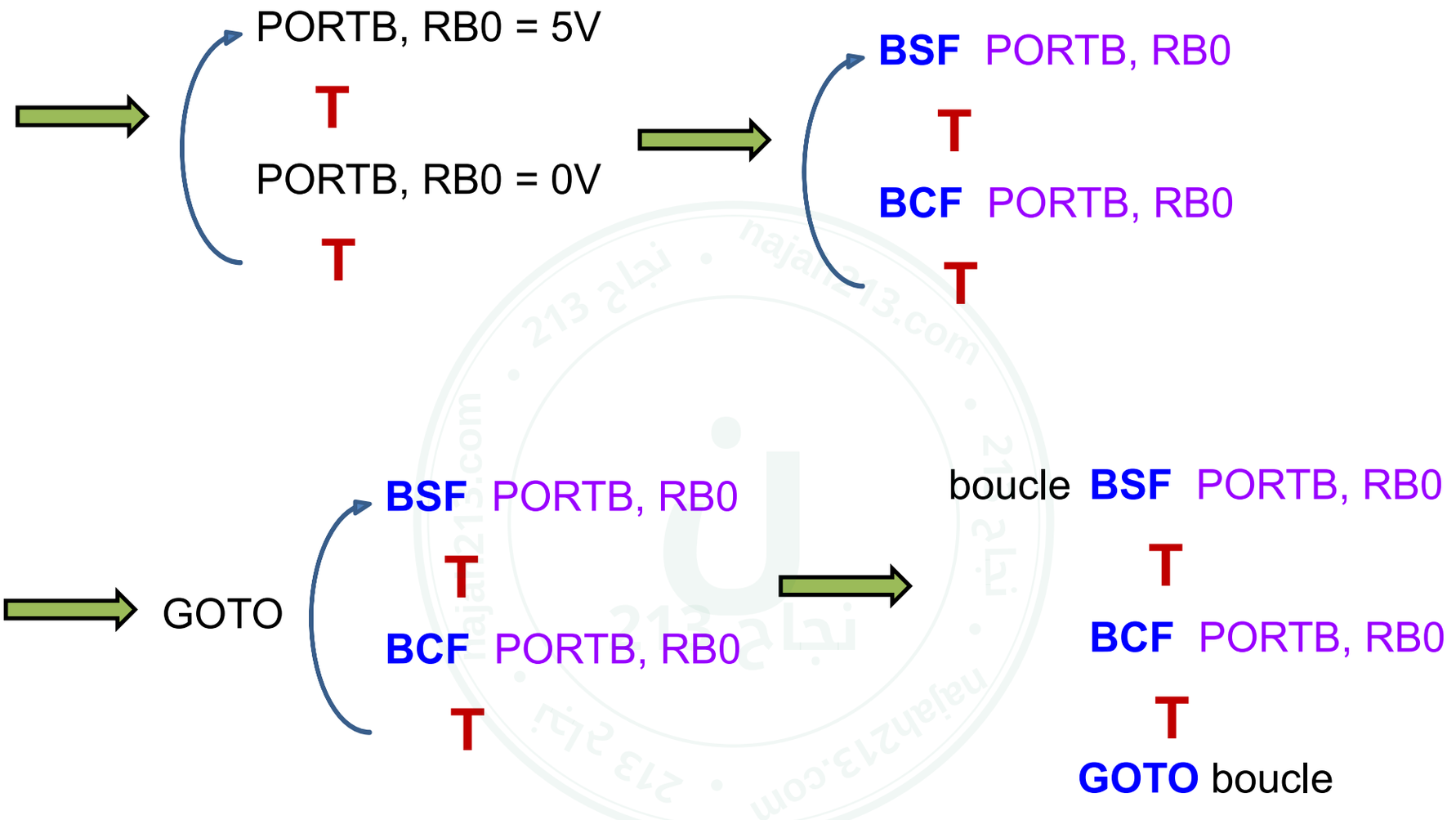
فروض واختبارات مع الحلول لكل المستويات — مجاناً







فروض واختبارات مع الحلول لكل المستويات — مجاناً



فروض واختبارات مع الحلول لكل المستويات — مجاناً

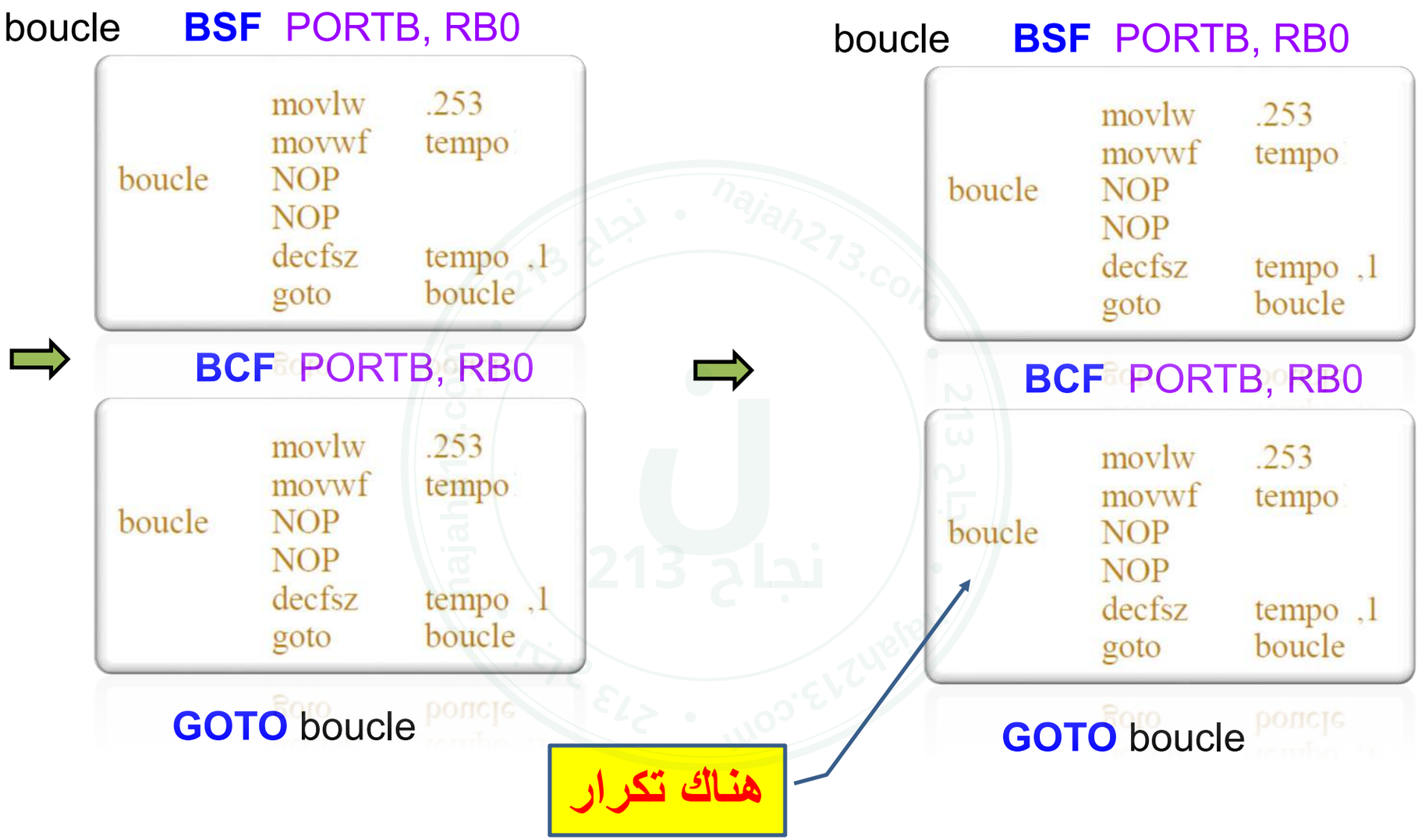
```
boucle  BSF  PORTB, RB0
        T
        BCF  PORTB, RB0
        T
        GOTO boucle
```

T= Tempo

```
boucle  BSF  PORTB, RB0
        Tempo
        BCF  PORTB, RB0
        Tempo
        GOTO boucle
```

Tempo ⇨

```
boucle  movlw   .253
        movwf   tempo
        NOP
        NOP
        decfsz  tempo ,1
        goto    boucle
```



فروض واختبارات مع الحلول لكل المستويات — مجاناً

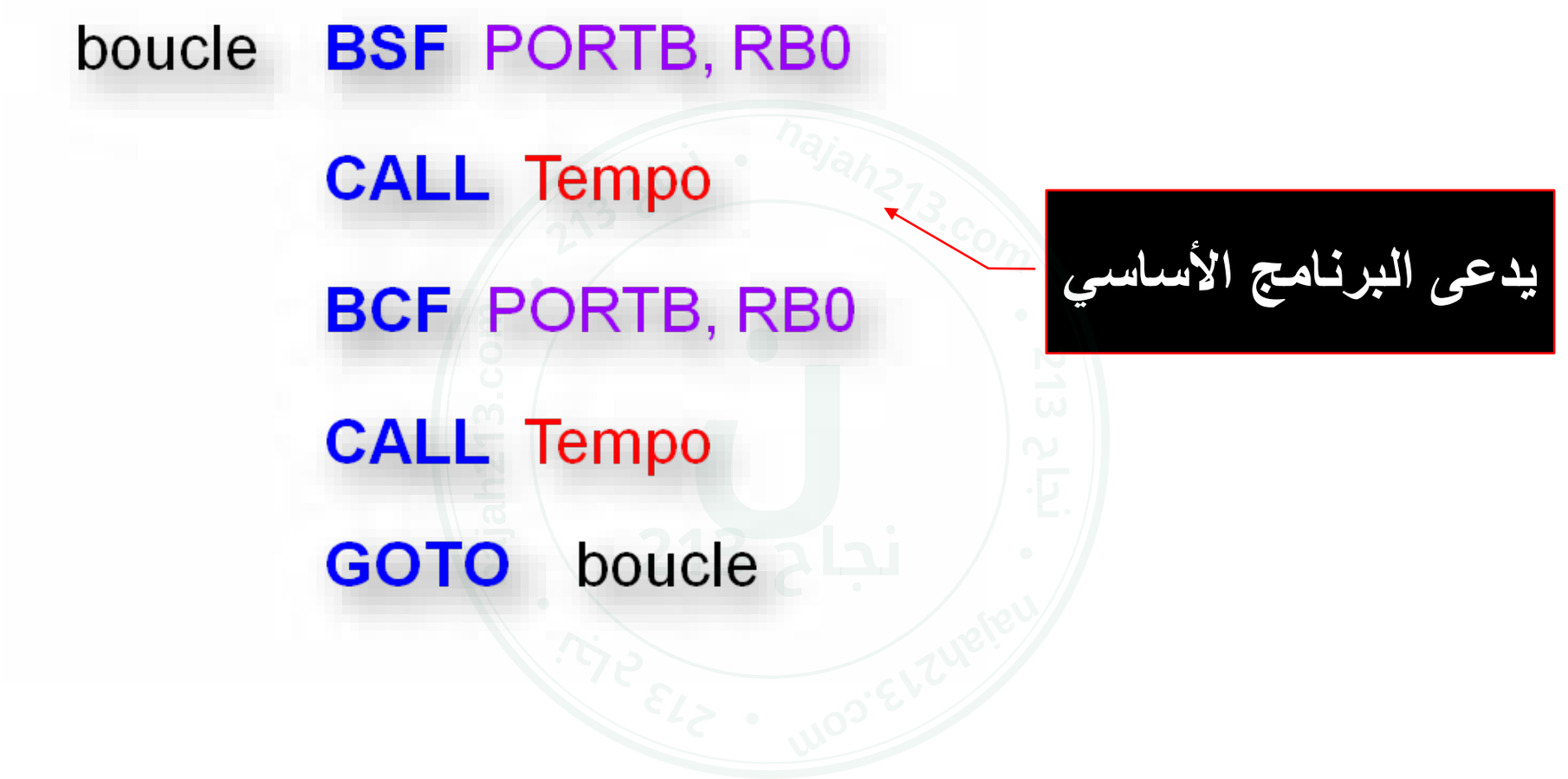
لتفادي التكرار نستعمل التعليمة
CALL

```
boucle   BSF  PORTB, RB0
         CALL Tempo
         BCF  PORTB, RB0
         CALL Tempo
         GOTO boucle

Tempo
         movlw   .253
         movwf   tempo
boucle   NOP
         NOP
         decfsz  tempo ,1
         goto    boucle
```

```
boucle   BSF  PORTB, RB0
         CALL Tempo
         BCF  PORTB, RB0
         CALL Tempo
         GOTO boucle

Tempo
         movlw   .253
         movwf   tempo
boucle   NOP
         NOP
         decfsz  tempo ,1
         goto    boucle
```

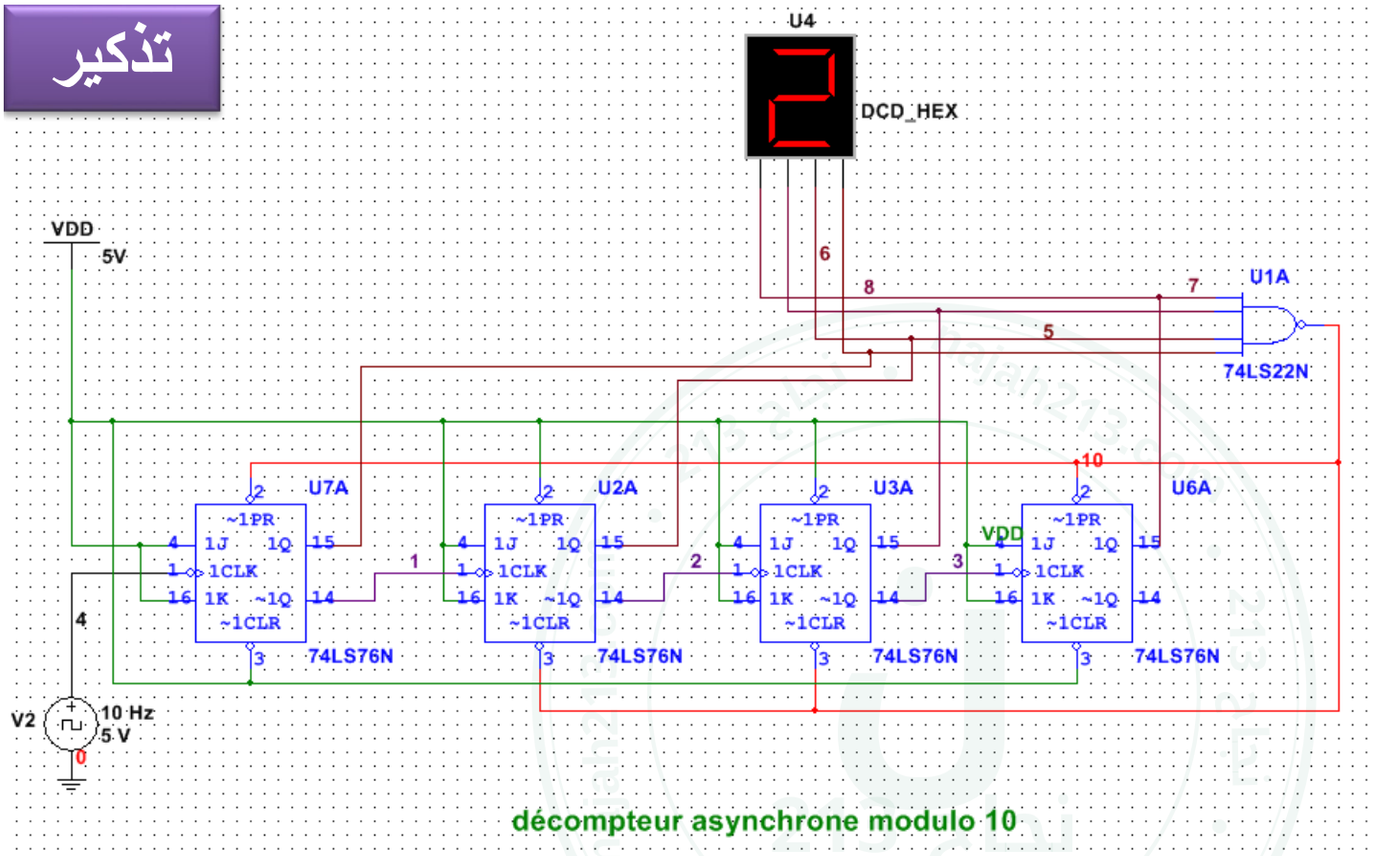


فروض واختبارات مع الحلول لكل المستويات — مجاناً



فروض واختبارات مع الحلول لكل المستويات — مجاناً

تذكير



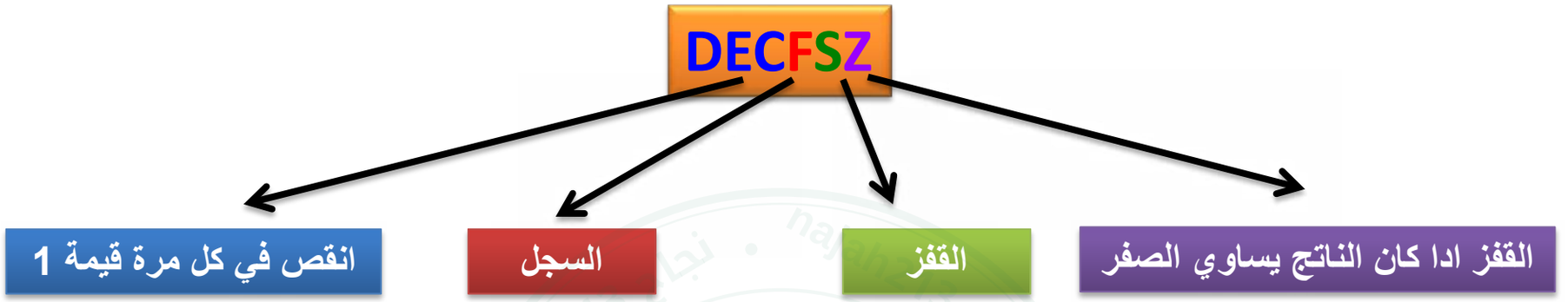
9	9
8	8
7	7
6	6
5	5
4	4
3	3
2	2
1	1
0	0

GOTO

العداد التنازلي في كل مرة ينقص القيمة 1

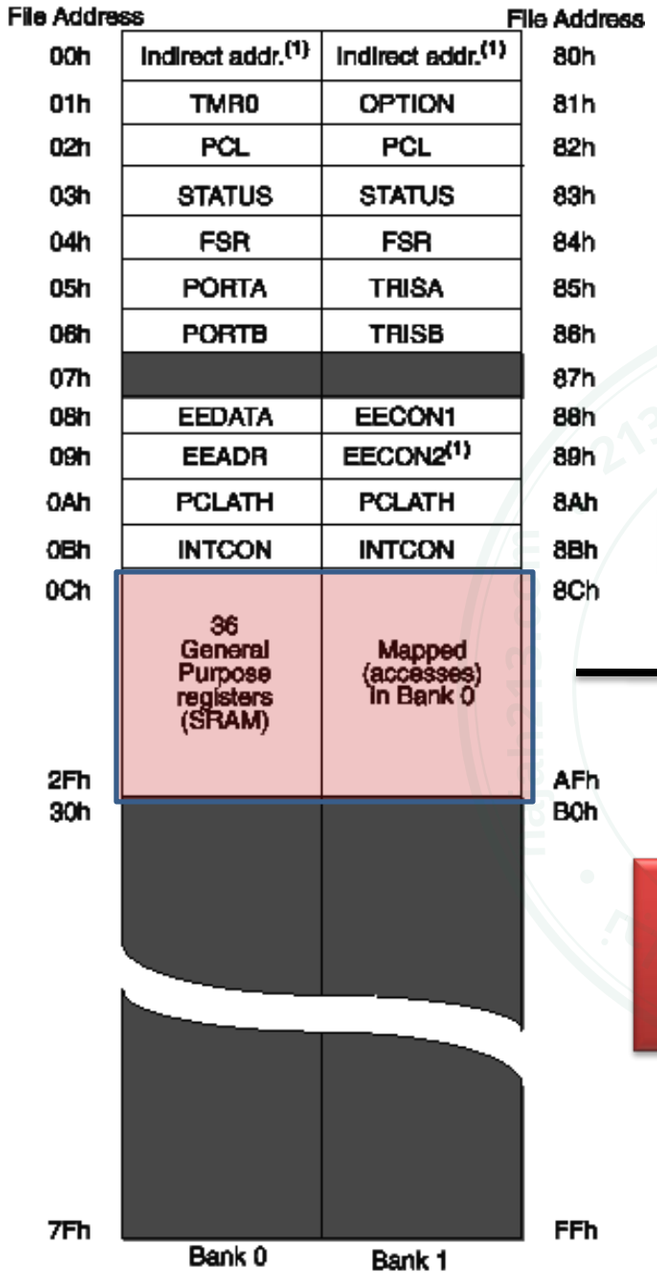
9-1=8	DECFSZ,1
8-1=7	DECFSZ,1
7-1=6	DECFSZ,1
6-1=5	DECFSZ,1
5-1=4	DECFSZ,1
4-1=3	DECFSZ,1
3-1=2	DECFSZ,1
2-1=1	DECFSZ,1
1-1=0	DECFSZ,1

عمل التعليمة decfsz له نفس عمل العداد التنازلي



فروض واختبارات مع الحلول لكل المستويات — مجاناً

الحلقات (Boucles)



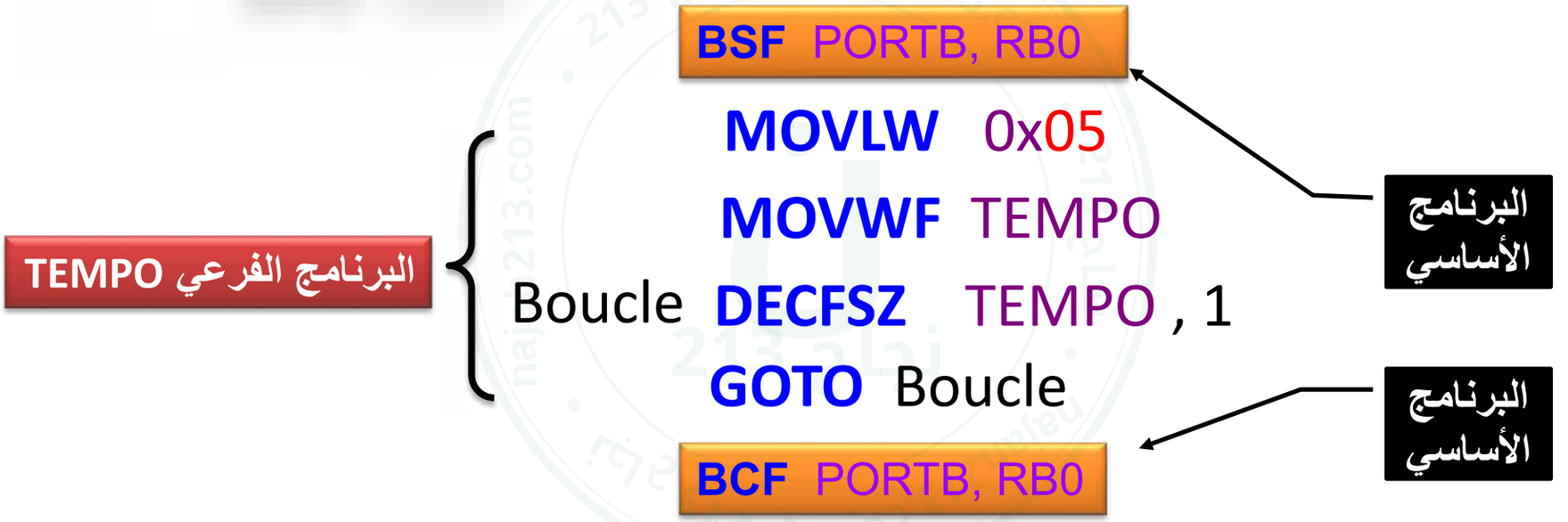
نقوم بإنشاء سجل TEMPO وإعطاءه عنوان مابين 0C و 2F

TEMPO

نقوم بشحن السجل TEMPO بالقيمة 0X05 على سبيل المثال ولسهولة الشرح.

الحلقات (Boucles)

```
boucle   BSF  PORTB, RB0
         CALL  Tempo
         BCF  PORTB, RB0
         CALL  Tempo
         GOTO   boucle
```



فروض واختبارات مع الحلول لكل المستويات — مجاناً

كيفية العمل

```
boucle  BSF  PORTB, RB0
        CALL  Tempo
        BCF  PORTB, RB0
        CALL  Tempo
        GOTO  boucle
```

BSF PORTB, RB0

MOVLW 0x05

MOVWF TEMPO

5-1=1

DECFSZ TEMPO , 1 + **GOTO**

4-1=3

DECFSZ TEMPO , 1 + **GOTO**

3-1=2

DECFSZ TEMPO , 1 + **GOTO**

2-1=1

DECFSZ TEMPO , 1 + **GOTO**

1-1=0

DECFSZ TEMPO , 1 + **SAUT**

BCF PORTB, RB0

فروض واختبارات مع الحلول لكل المستويات — مجاناً

كيفية العمل

```
MOVLW  0x05          → (1us)
MOVWF  TEMPO         → (1us)
Boucle DECFSZ  TEMPO , 1 → (1(2)us)
       GOTO  Boucle  → (2us)
```

5-1=1 → DECFSZ+ GOTO →**(1+2) us**

4-1=3 → DECFSZ+ GOTO →**(1+2) us**

3-1=2 → DECFSZ+ GOTO →**(1+2) us**

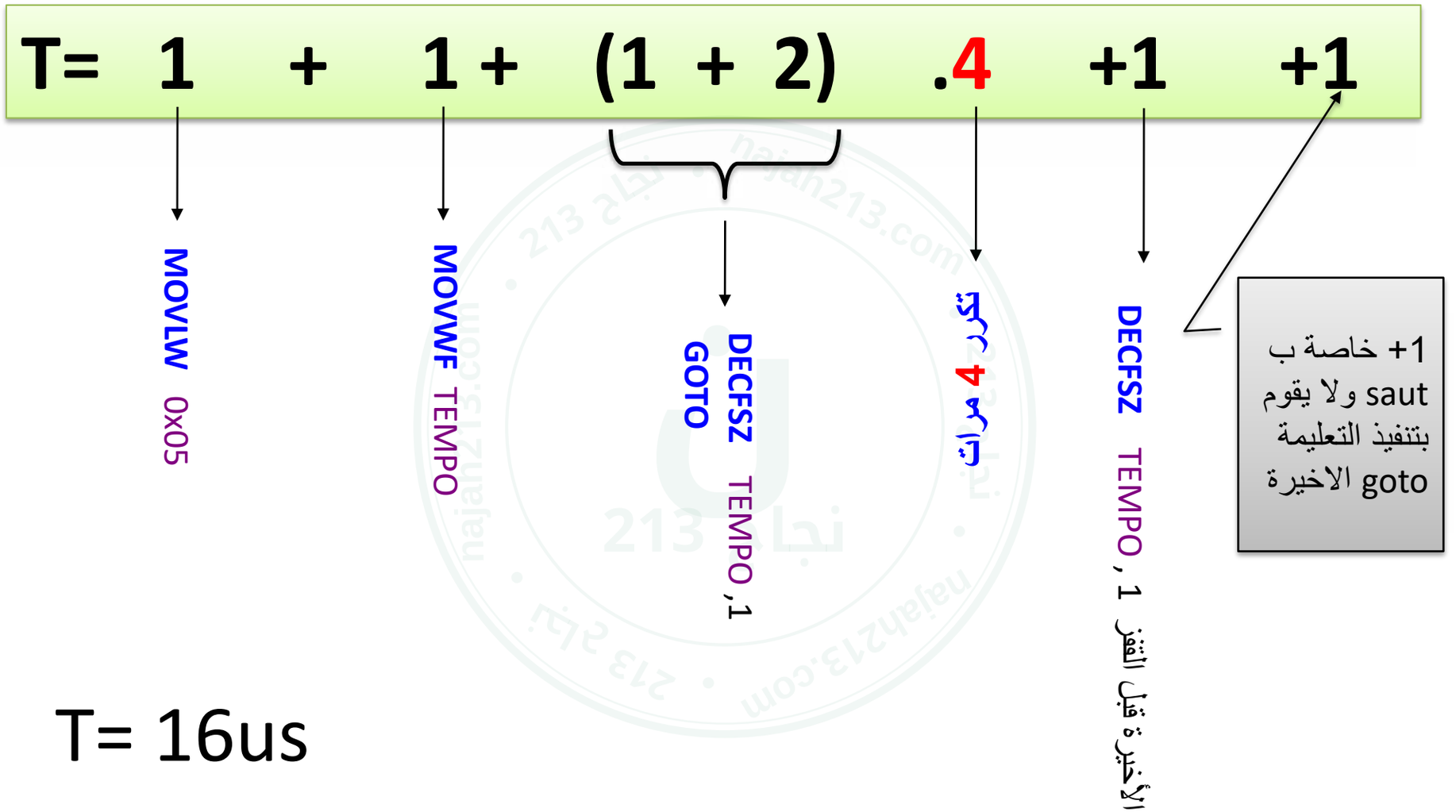
2-1=1 → DECFSZ+ GOTO →**(1+2) us**

1-1=0 → DECFSZ+ **SAUT** →**(1+1) us**

ملاحظة: عند العملية 1-1=0 فان ميكرومراقب يقوم بتنفيذ التعليمة Decfsz ويقوم بعدها بالقفز اي انه لاينفذ التعليمة goto

زمن التأجيل

$$T= 1+1+(1+2).4+1+1$$



$$T= 16us$$

ملاحظة: 4=5-1

تعميم

```
        MOVLW   0xN          → (1us)
        MOVWF   TEMPO        → (1us)
Boucle  DECFSZ  TEMPO , 1    → (1(2)us)
        GOTO    Boucle       → (2us)
```

$$T= 1+1+(1+2).(N-1)+1+1$$

اكبر قيمة لـ N هي 256 وهي القيمة القصوى للسجلات مشفرة على 8 بيت

T= 16us

هذا الزمن صغير جدا اذا ما قرن 1s (1s=1000000us)

الحل: هو إضافة التعليمة NOP داخل الحلقة

```
        MOVLW   0x05          → (1us)
        MOVWF   TEMPO         → (1us)
Boucle  NOP
        DECFSZ  TEMPO , 1     → (1(2)us)
        GOTO    Boucle        → (2us)
```

+1 خاصة بالتعليمة NOP

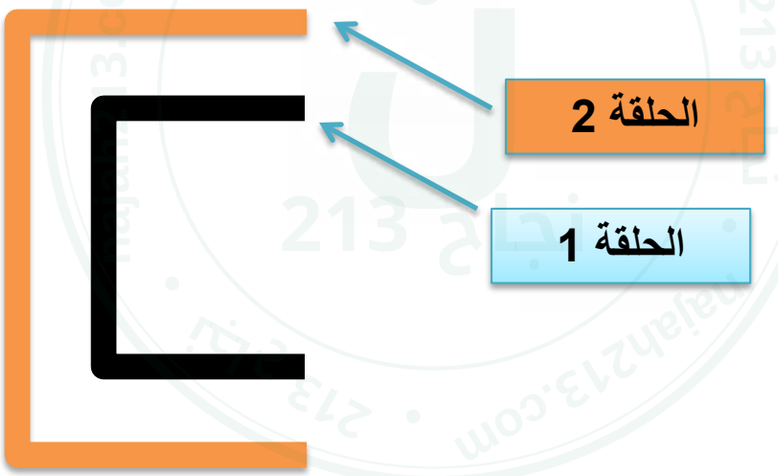
$$T= 1+1+(1+1+2).4+1+1$$

$$T= 20us$$

T= 20us

هذا الزمن صغير جدا اذا ما قرن 1s (1s=1000000us)

الحل: هو استعمال الحلقات المتداخلة

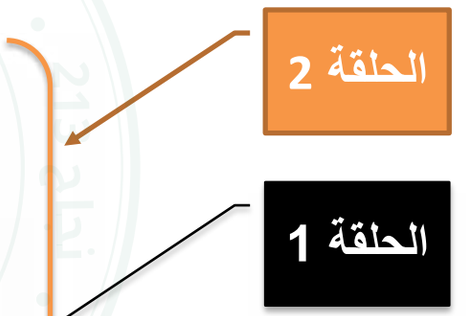


شكل الحلقات المتداخلة

نقوم بإنشاء سجل TEMP1 (الحلقة 1) و TEMP2 (الحلقة2) وإعطاءهما عنوان مابين 0C و 2F

```
boucle   BSF  PORTB, RB0
         CALL  Tempo
         BCF  PORTB, RB0
         CALL  Tempo
         GOTO   boucle
```

```
Tempo    MOVLW   0x07
         MOVWF   TEMP2
Boucle2  MOVLW   0x05
         MOVWF   TEMP1
Boucle1  DECFSZ   TEMP1 , 1
         GOTO  Boucle1
         DECFSZ   TEMP2 , 1
         GOTO  Boucle2
```



كيفية العمل



وهكذا تتكرر العملية

لسهولة الدراسة نقوم بوضع هذه الحلقة(الحلقة1) كأنها تكافئ تعليمة ما تستغرق 16us

```
Boucle2   MOVLW   0x05
          MOVWF   TEMP1
Boucle1   DECFSZ  TEMP1 , 1
          GOTO    Boucle1
```

ولتكن هذه التعليمة مثلا NOP



```
        MOVLW   0x07
        MOVWF   TEMP2
Boucle2 MOVLW   0x05
        MOVWF   TEMP1
Boucle1 DECFSZ  TEMP1 , 1
        GOTO    Boucle1
        DECFSZ  TEMP2 , 1
        GOTO    Boucle2
```

NOP(16us)



```
        MOVLW   0x07
        MOVWF   TEMP2
Boucle2 NOP(16us)
        DECFSZ  TEMP2 , 1
        GOTO    Boucle2
```

كيفية العمل

```
                    7-1=6
DECFSZ   TEMP2 , 1 + GOTO
                    6-1=5
DECFSZ   TEMP2 , 1 + GOTO
                    5-1=4
DECFSZ   TEMP2 , 1 + GOTO
                    4-1=3
DECFSZ   TEMP2 , 1 + GOTO
  ---------------------------------
  ---------------------------------
  ---------------------------------
                    1-1=0
DECFSZ   TEMP2 , 1 + SAUT
BCF  PORTB,RB0
```

7-1=1 → DECFSZ+ GOTO →**(16+1+2) us**

6-1=5 → DECFSZ+ GOTO →**(16+1+2) us**

5-1=4 → DECFSZ+ GOTO →**(16+1+2) us**

4-1=1 → DECFSZ+ GOTO →**(16+1+2) us**

1-1=0 → DECFSZ+ **SAUT** →**(16+1+1) us**

فروض واختبارات مع الحلول لكل المستويات — مجاناً

```
MOVLW   0x07  →1us
MOVWF   TEMP2 → 1us
Boucle2 NOP(16us)
DECFSZ  TEMP2 , 1 → 1(2)us
GOTO    Boucle2 → 2us
```

$$T= 1+1+(16+1+2).6+16+1+1$$

$$T= 134us$$

تعميم

$$T= 1+1+(16+1+2).(N2-1)+16+1+1$$

تعميم

سوف تعوض 16us ← T= 1+1+(1+2).(N1-1)+1+1

$$T= 1+1+(16+1+2).(N2-1)+16+1+1$$

T= 1+1+[1+1+(1+2).(N-1)+1+1 +1+2].(N2-1)+ 1+1+(1+2).(N1-1)+1+1 +1+1

T= 1+1+[1+1+(1+2).(N1-1)+1+1 +1+2].(N2-1)+ 1+1+(1+2).(N1-1)+1+1 +1+1

تطبيقات

```
        MOVLW   0x256
        MOVWF   TEMPO
Boucle  DECFSZ  TEMPO , 1
        GOTO    Boucle
```

T=

فروض واختبارات مع الحلول لكل المستويات — مجاناً

تطبيقات

```
            movlw     .253
            movwf     tempo1
boucle1     NOP
            NOP
            decfsz    tempo1,1
            goto      boucle1
```

T=

T= $1266\mu s = 1{,}266ms$

```
1     tempo      movlw     .197
2                movwf     tempo0
3     boucle     movlw     .253
4                movwf     tempo1
5     boucle1    NOP
6                NOP
7                decfsz    tempo1,1
8                goto      boucle1
9                decfsz    tempo0,1
10               goto      boucle
11               NOP
12               NOP
13               return
```

T= 249 998µs= 249,998µs

فروض واختبارات مع الحلول لكل المستويات — مجاناً

Programme final

```
1            LIST     p=16f84
2            #include "p16f84.inc"
3            __config 0x3FF2
4   ;
5   tempo0   EQU      0x0C
6   tempo1   EQU      0x0D
7   ;
8            org      0x00
9            bsf      STATUS,RP0
10           movlw    0xFE
11           movwf    TRISB
12           bcf      STATUS,RP0
13  start
14           bsf      PORTB,0
15           call     tempo
16           bcf      PORTB,0
17           call     tempo
18           goto     start
19  ;
20  tempo    movlw    .197
21           movwf    tempo0
22  boucle   movlw    .253
23           movwf    tempo1
24  boucle1  NOP
25           NOP
26           decfsz   tempo1,1
27           goto     boucle1
28           decfsz   tempo0,1
29           goto     boucle
30           NOP
31           NOP
32           return
33  ;
34           end
```

T= 249 998µs= 249,998µs

```
Tempo   movlw 0xF9
        movwf compt
        NOP
Boucle  NOP
        decfsz compt,1
        goto boucle
        return
```

Re : temporisation (PIC 16F84A)

merci pour votre réponse

oui $(F9)_{16} = (249)_{10}$ mais ce que j'ai écrit c'est seulement pour 1ms d'aprés mon programme: tempo=3+(1+1+2)248+(1+2)+2=100 0 cycle donc 1000cycle *1µs=1ms non??

فروض واختبارات مع الحلول لكل المستويات — مجاناً