Fetch Cycle

指令周期 (instruction cycle)

- **指令周期**，又稱「**提取－執行周期**」（fetch-and-execute cycle）是指CPU要執行一條「**機器指令**」經過的步驟，由若干**機器周期**組成。

- 執行一條「**機器指令**」經過下面兩個週期「**提取周期**」**與**「**執行周期**」

- 「**提取周期**」**(Fetch Cycle)**
  - ✓ 取得指令：CPU內有「程式計數器」（Program Counter），它儲存下一個要執行的指令的地址。處理器按PC儲存的地址，經主記憶體取得指令的內容，並將指令存入指令暫存器（IR）。
  - ✓ 解碼指令：將指令暫存器（IR）內的指令譯成機器語言。

- 「**執行周期**」**(Execution Cycle)**
  - ✓ 執行指令
  - ✓ 儲存結果

- Ref. https://zh.wikipedia.org/wiki/%E6%8C%87%E4%BB%A4%E5%91%A8%E6%9C%9F

Assembly

```
    ;list p=16F1826
#include    <p16Lf1826.inc>

temp1       equ h'20'
temp2       equ h'21'
temp        equ 0x25
;
;*************************************
;           Program start             *
;*************************************
            org         0x00

            movlw       0x34
            movwf       temp1
            movlw       0x99
            movwf       PORTB
            nop         ;cpfsgteq       temp1
            movlw       0x66
            movwf       PORTB

            movlw       0xaa
            movwf       temp1
            movlw       0x99
            nop         ;cpfsgteq       temp1
            movlw       0x77
            movwf       PORTB

            movlw       0x34
            movwf       temp1
            movlw       0x99
            nop         ;cpfslt         temp1
            movlw       0x88
            movwf       PORTB

            movlw       0xaa
            movwf       temp1
            movlw       0x99
            nop         ;cpfslt         temp1
            movlw       0x55
            movwf       PORTB

            goto        $

            end
```

Assembly => machine code

Address | machine code | Assembly

```
00000020            00004 temp1           equ     h'20'
00000021            00005 temp2           equ     h'21'
00000025            00006 temp            equ     0x25
                    00007 ;
                    00008 ;*****************************************
                    00009 ;           Program start                *
                    00010 ;*****************************************
                    00011                         org             0x00
                    00012
0000  3034          00013                         movlw           0x34
0001  00A0          00014                         movwf           temp1
0002  3099          00015                         movlw           0x99
0003  008D          00016                         movwf           PORTB
0004  0000          00017                         nop                     ;cpfsgteq       temp1
0005  3066          00018                         movlw           0x66
0006  008D          00019                         movwf           PORTB
                    00020
0007  30AA          00021                         movlw           0xaa
0008  00A0          00022                         movwf           temp1
0009  3099          00023                         movlw           0x99
000A  0000          00024                         nop                     ;cpfsgteq       temp1
000B  3077          00025                         movlw           0x77
000C  008D          00026                         movwf           PORTB
                    00027
000D  3034          00028                         movlw           0x34
000E  00A0          00029                         movwf           temp1
000F  3099          00030                         movlw           0x99
0010  0000          00031                         nop                     ;cpfslt                 temp1
0011  3088          00032                         movlw           0x88
0012  008D          00033                         movwf           PORTB
                    00034
0013  30AA          00035                         movlw           0xaa
0014  00A0          00036                         movwf           temp1
0015  3099          00037                         movlw           0x99
0016  0000          00038                         nop                     ;cpfslt                 temp1
0017  3055          00039                         movlw           0x55
0018  008D          00040                         movwf           PORTB
                    00041
0019  2???          00042                         goto            $
```

PIC16F1826 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSB			LSB	Status Affected	Notes
LITERAL OPERATIONS									
ADDLW	k	Add literal and W	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k	kkkk		
MOVLP	k	Move literal to PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	Move literal to W	1	11	0000	kkkk	kkkk		
SUBLW	k	Subtract W from literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note

1：If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

2：If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

指令資料流

49個指令分成八個類別，
從八個類別中各挑出部分指令做控制訊號及資料流向範例。

各指令執行所需時間不盡相同，大致上可由類別區分：

- 一個時間週期：
  Literal Operations、Inherent Operations。

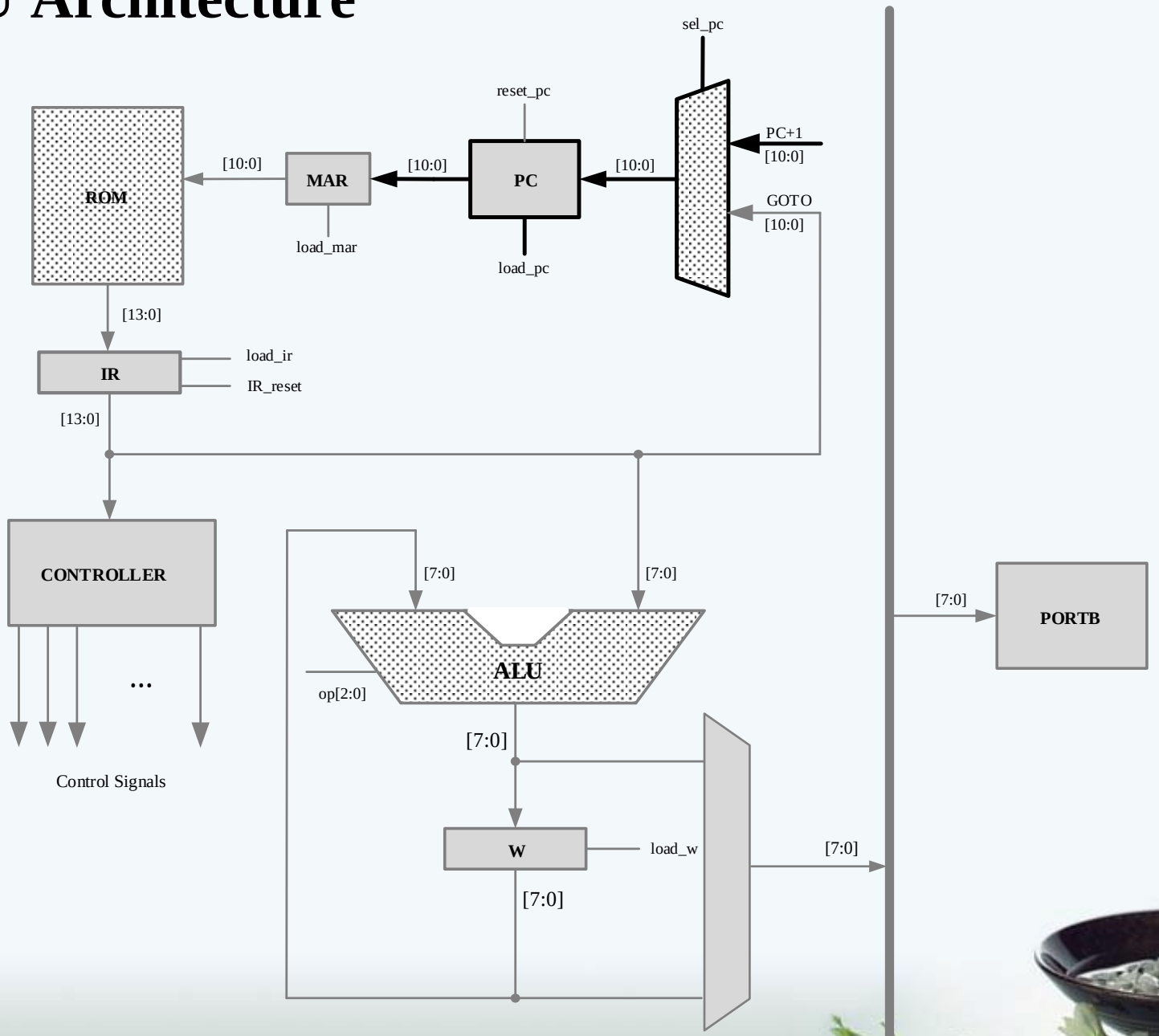
- 兩個時間週期：
  Byte-oriented File Register Operations、Bit-oriented File Register Operations、
  Bit-oriented Skip Operations。

- 三個時間週期：
  Byte-oriented Skip Operations、Control Operations、C-Compiler Optimized。

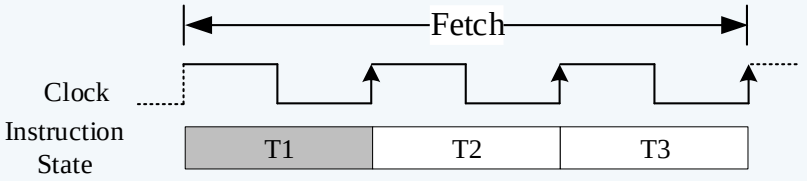
T_1、T_2及T_3擷取階段，控制訊號均相同，如下表所示。

狀態	動作	控制訊號
T_1	MAR←PC	load_mar
T_2	PC←PC+1	sel_pc; load_pc
T_3	IR←ROM[MAR]	load_ir

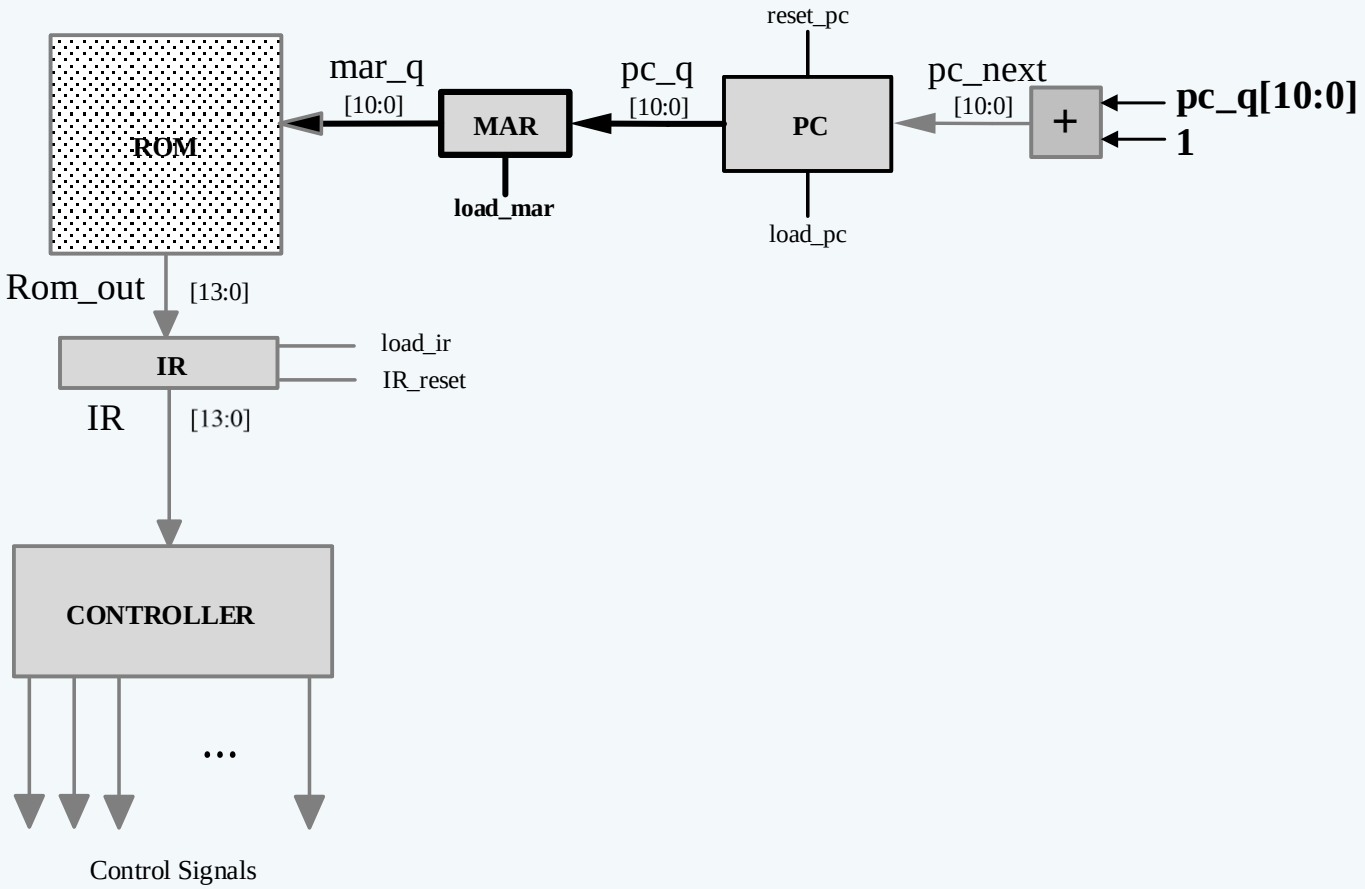
PIC MCU Architecture



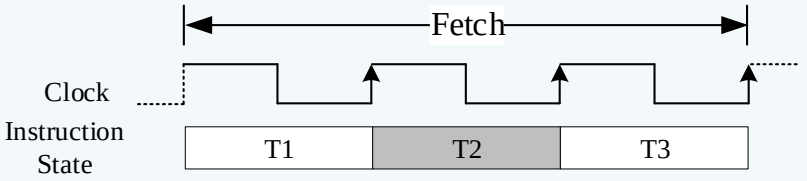
Fetch T_1



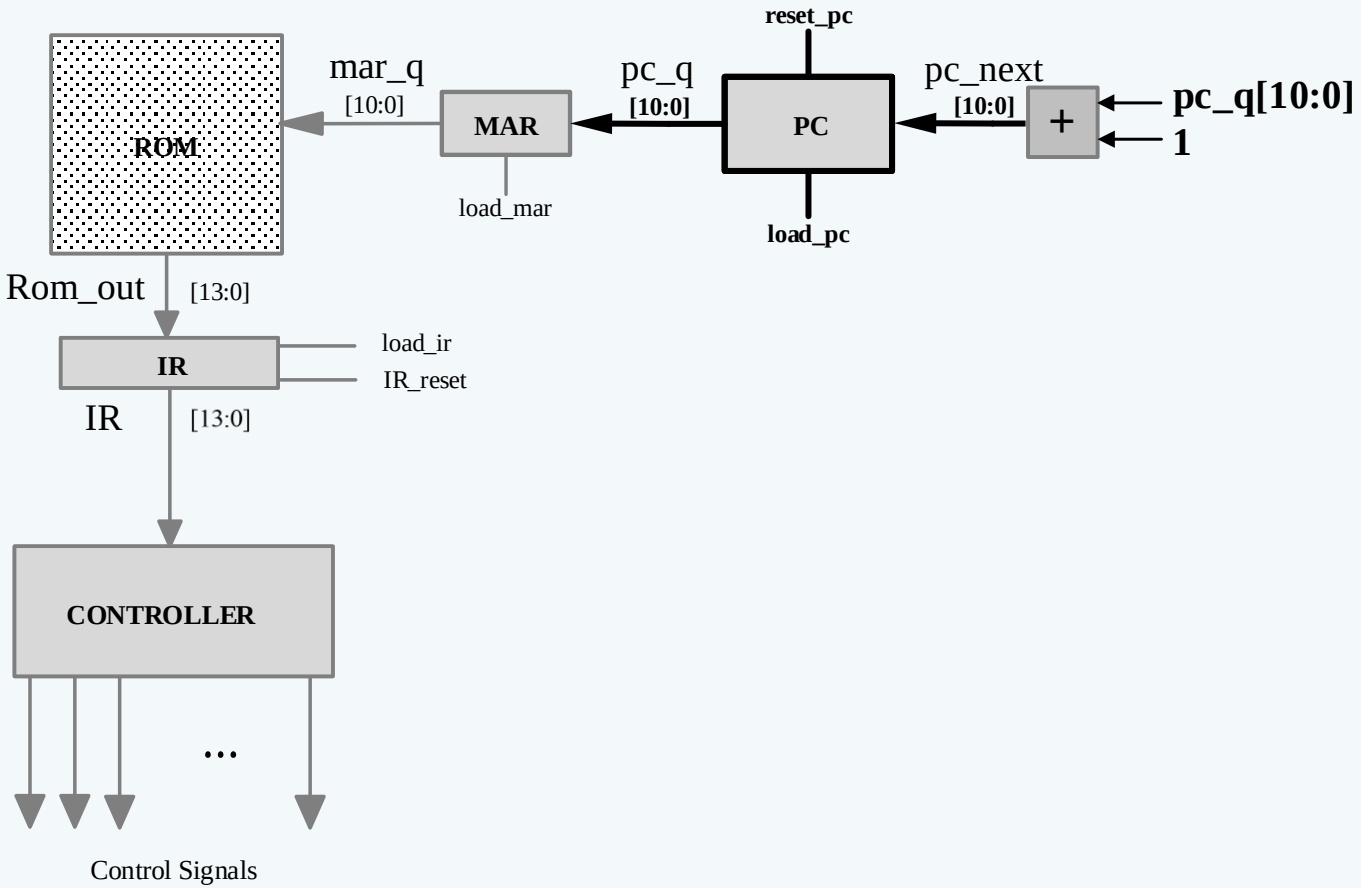
狀態	u-operation	控制訊號
T_1	MAR←PC	load_mar



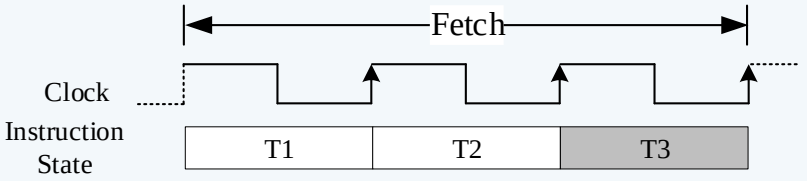
Fetch T_2



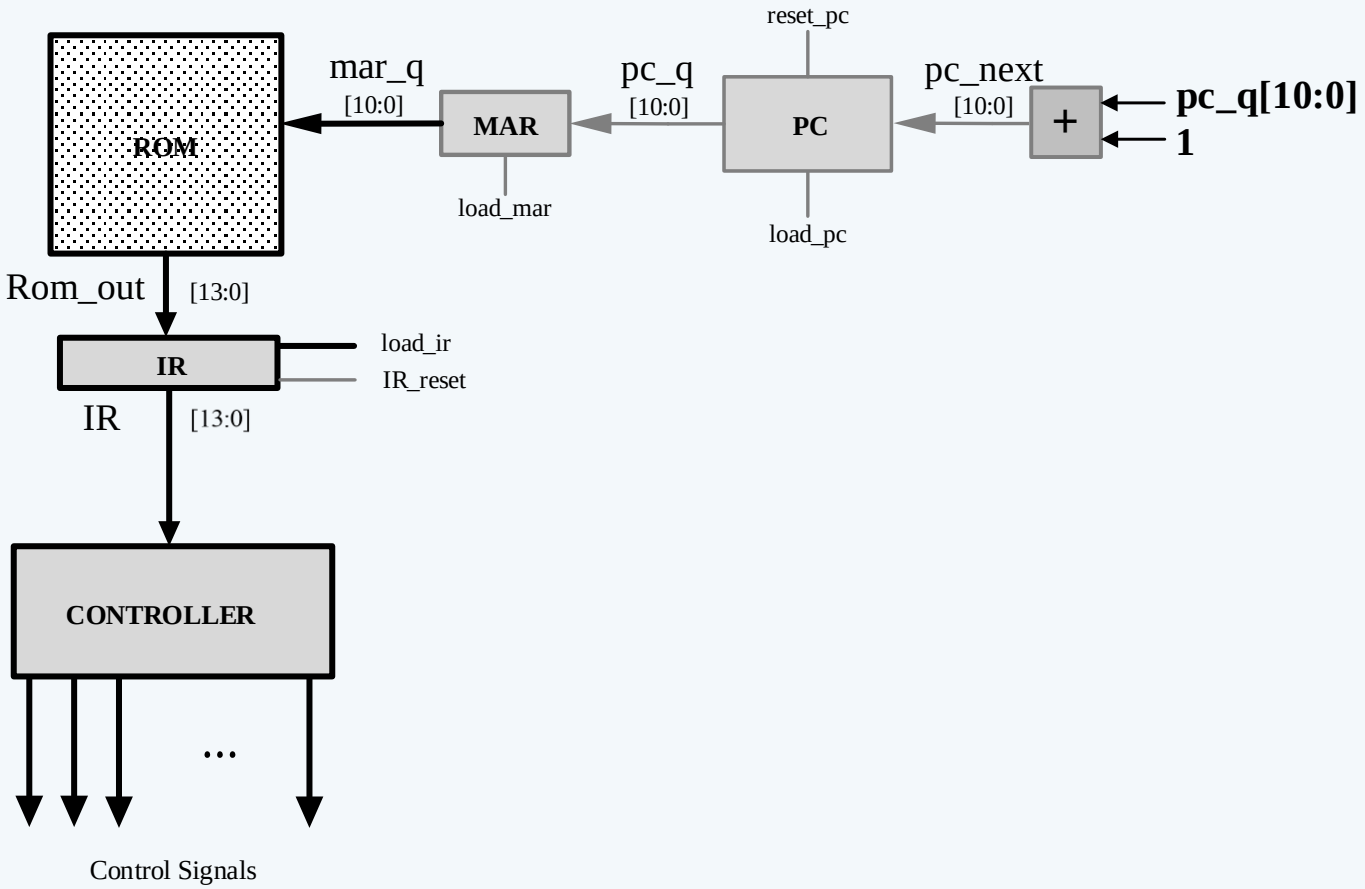
狀態	u-operation	控制訊號
T_2	PC←PC+1	load_pc



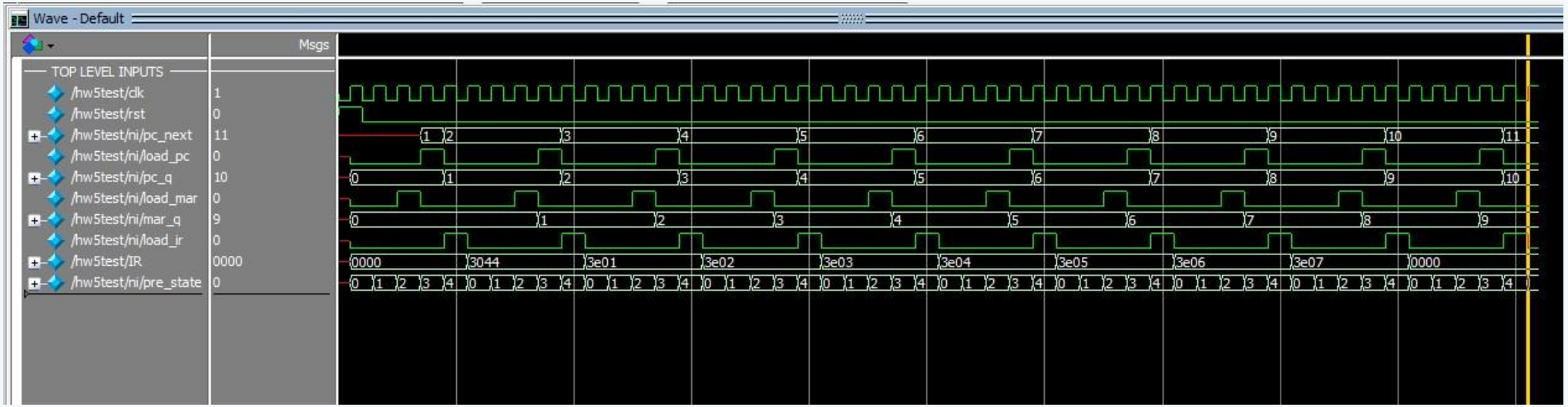
Fetch T_3



狀態	u-operation	控制訊號
T_3	IR←ROM[MAR]	load_ir



Fetch Cycle Waves



LITERAL OPERATIONS

PIC16F1826 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSB			LSB	Status Affected	Notes
LITERAL OPERATIONS									
ADDLW	k	Add literal and W	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k	kkkk		
MOVLP	k	Move literal to PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	Move literal to W	1	11	0000	kkkk	kkkk		
SUBLW	k	Subtract W from literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note

1：If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

2：If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

PIC16F1826 INSTRUCTION SET

OPCODE FIELD DESCRIPTIONS	
Field	**Description**
f	Register file address（0x00 to 0x7F）
W	Working register（accumulator）
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don’t care location（= 0 or 1） The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select： d = 0：store result in W d = 1：store result in file register f
n	FSR or INDF number（0-1）
mm	Pre-post increment-decrement mode selection

ABBREVIATION DESCRIPTIONS	
Field	**Description**
PC	Program Counter
TO	Time-out bit
$\overline{C}$	Carry bit
DC	Digit carry bit
Z	Zero bit
$\overline{PD}$	Power-down bit