立即數定址

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。

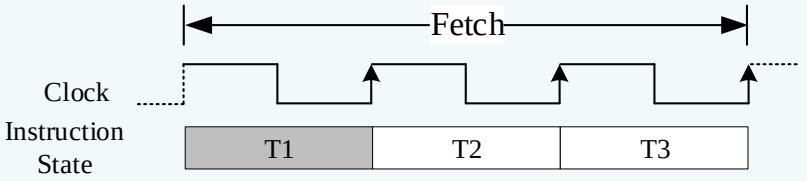
- 兩個時間週期：
  Byte-oriented File Register Operations、Bit-oriented File Register Operations、
  Bit-oriented Skip Operations。

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

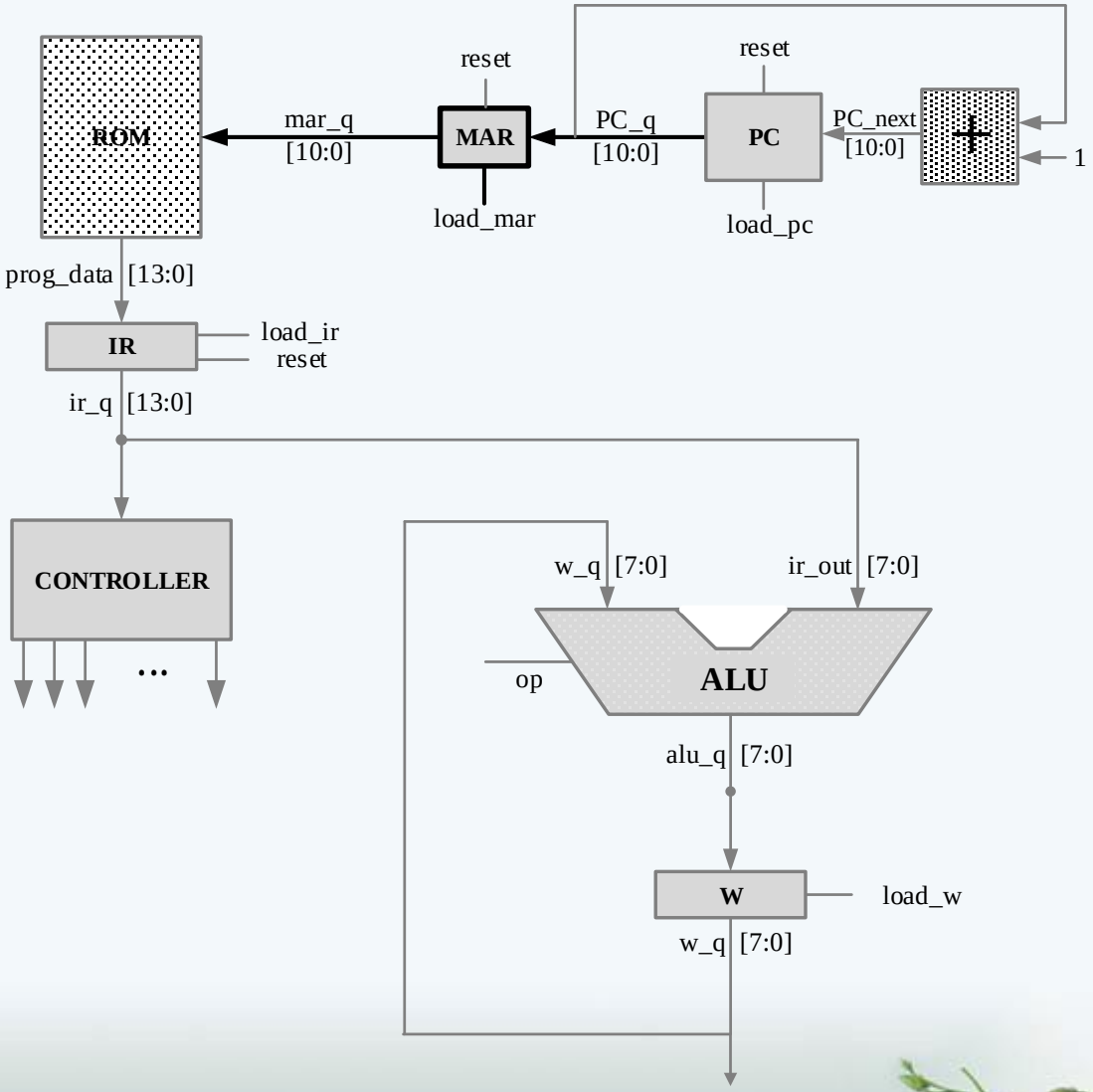
T_1、T_2及T_3擷取階段，控制訊號均相同，如下表所示。

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

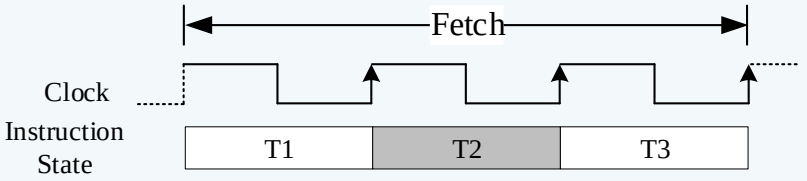
Fetch T1



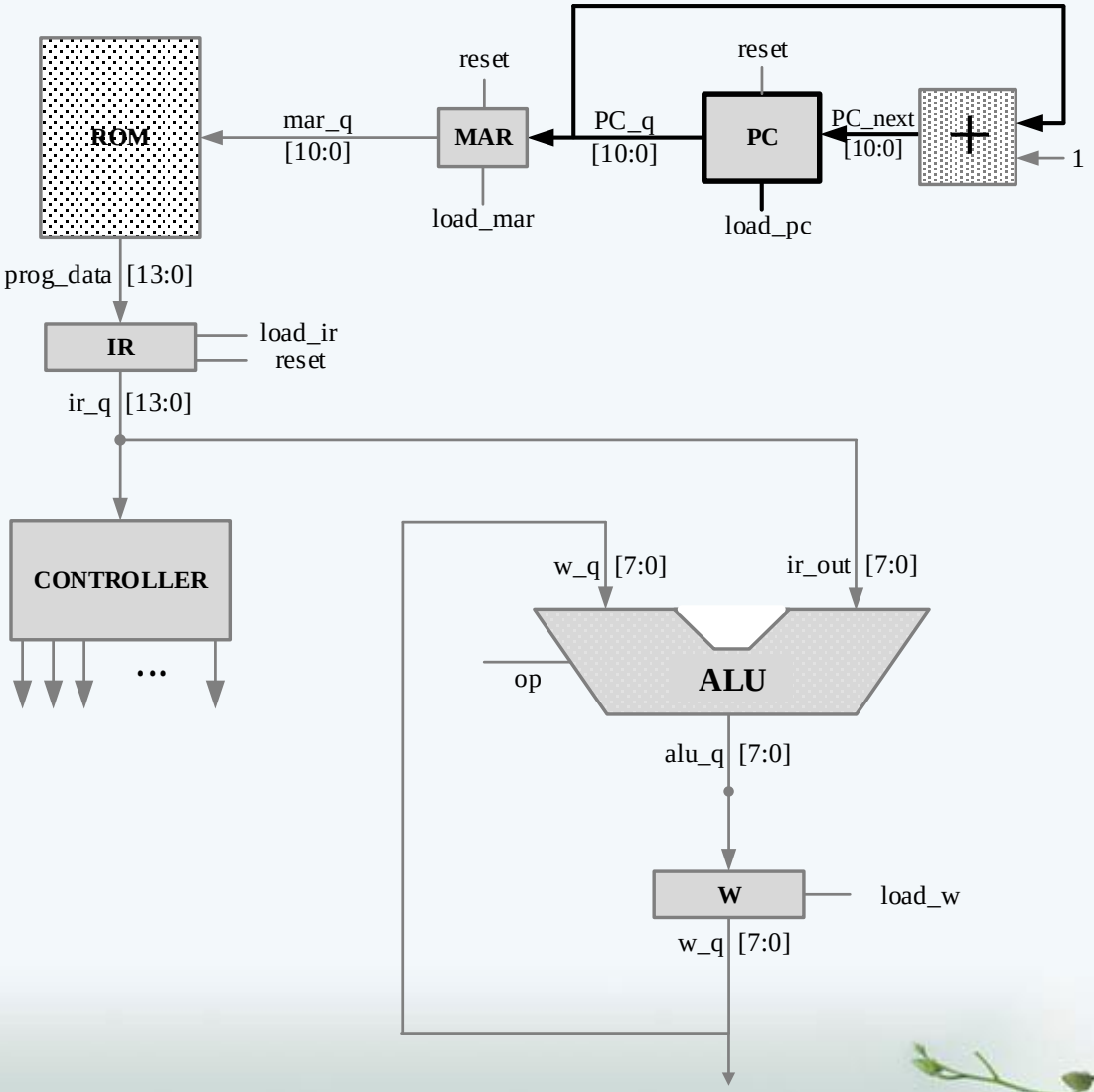
狀態	動作	控制訊號
T_1	MAR←PC	load_mar



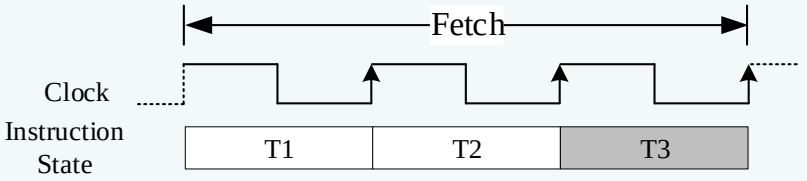
Fetch T2



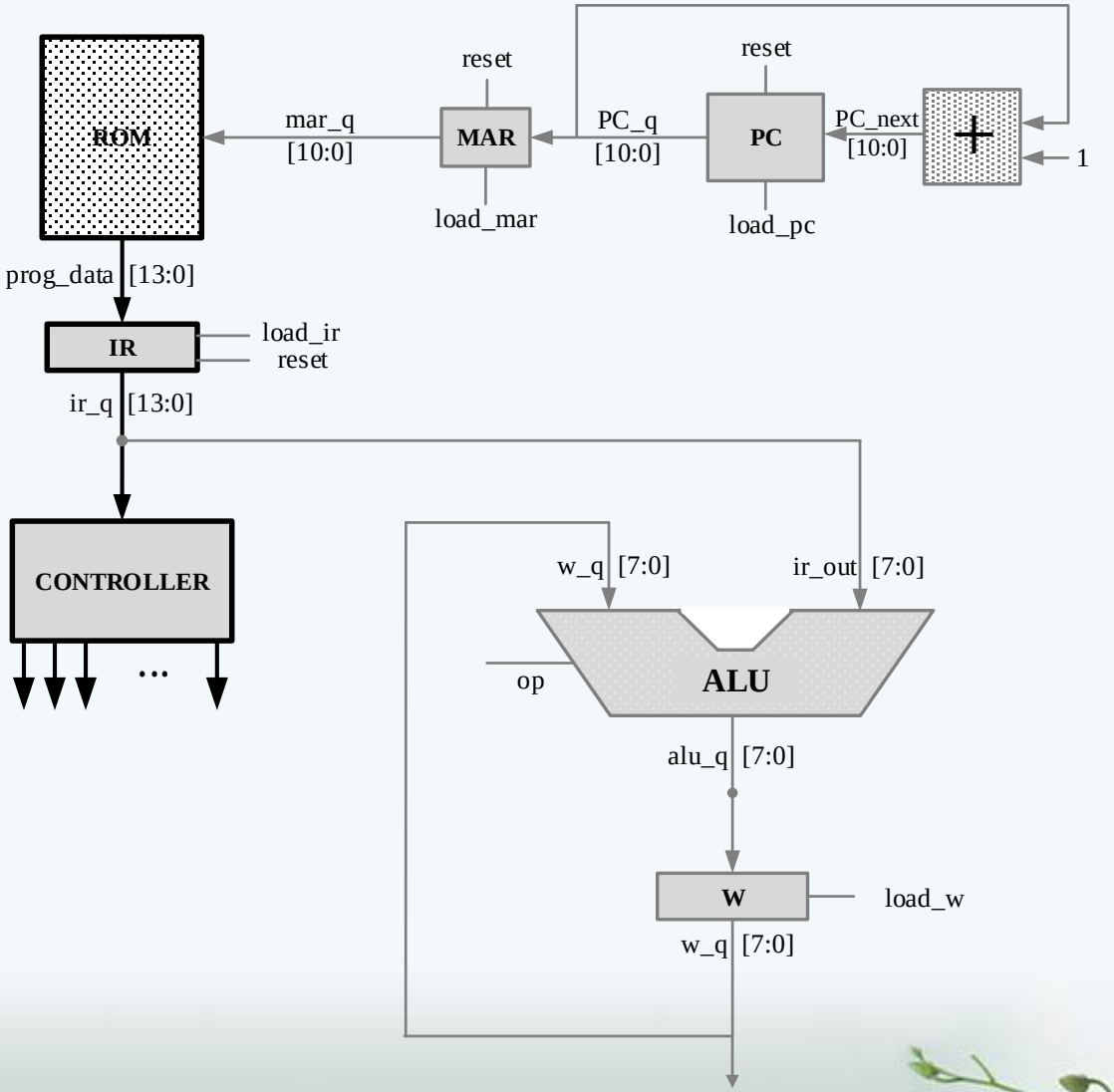
狀態	動作	控制訊號
T_2	PC←PC+1	load_pc



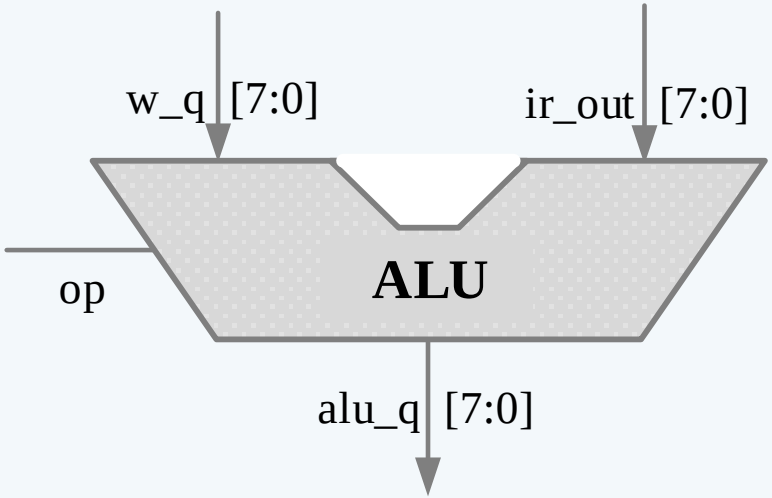
Fetch T3



狀態	動作	控制訊號
T_3	IR←ROM[MAR]	load_ir



ALU




```
always_comb
begin
    case(op)
        0: alu_q = ir_q[7:0] + w_q;
        1: alu_q = ir_q[7:0] - w_q;
        2: alu_q = ir_q[7:0] & w_q;
        3: alu_q = ir_q[7:0] | w_q;
        4: alu_q = ir_q[7:0] ^ w_q;
        5: alu_q = ir_q[7:0];
        default: alu_q = ir_q[7:0] + w_q;
    endcase
end
```

PIC16F1826 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSB LSB	Status Affected	Notes
LITERAL OPERATIONS						
MOVLW	k	Move literal to W	1	11 0000 kkkk kkkk		
ADDLW	k	Add literal and W	1	11 1110 kkkk kkkk	C, DC, Z	
SUBLW	k	Subtract W from literal	1	11 1100 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	
XORLW	k	Exclusive OR literal with W	1	11 1010 kkkk kkkk	Z	

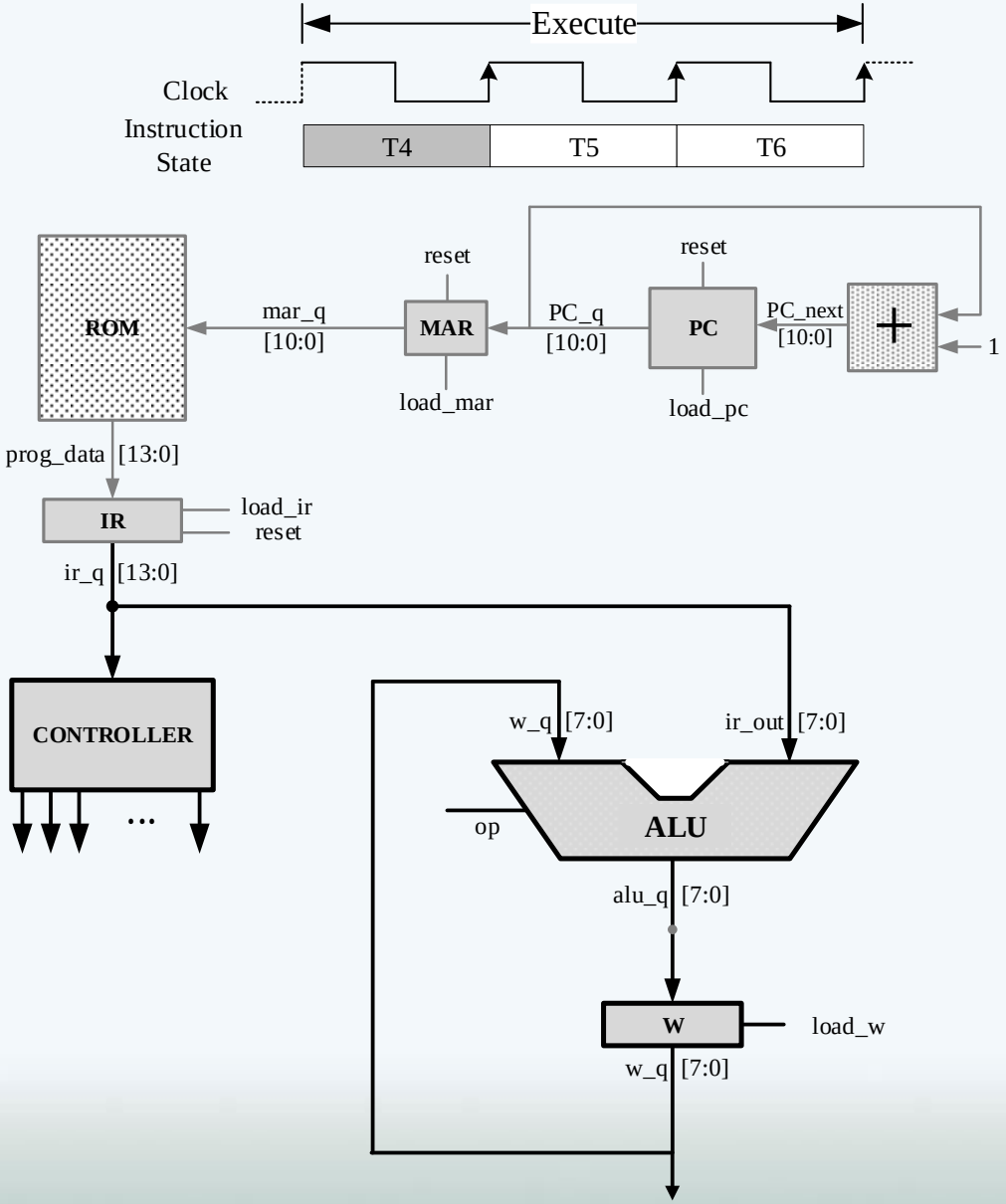
Note

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.

MOVLW：將立即數傳送到W

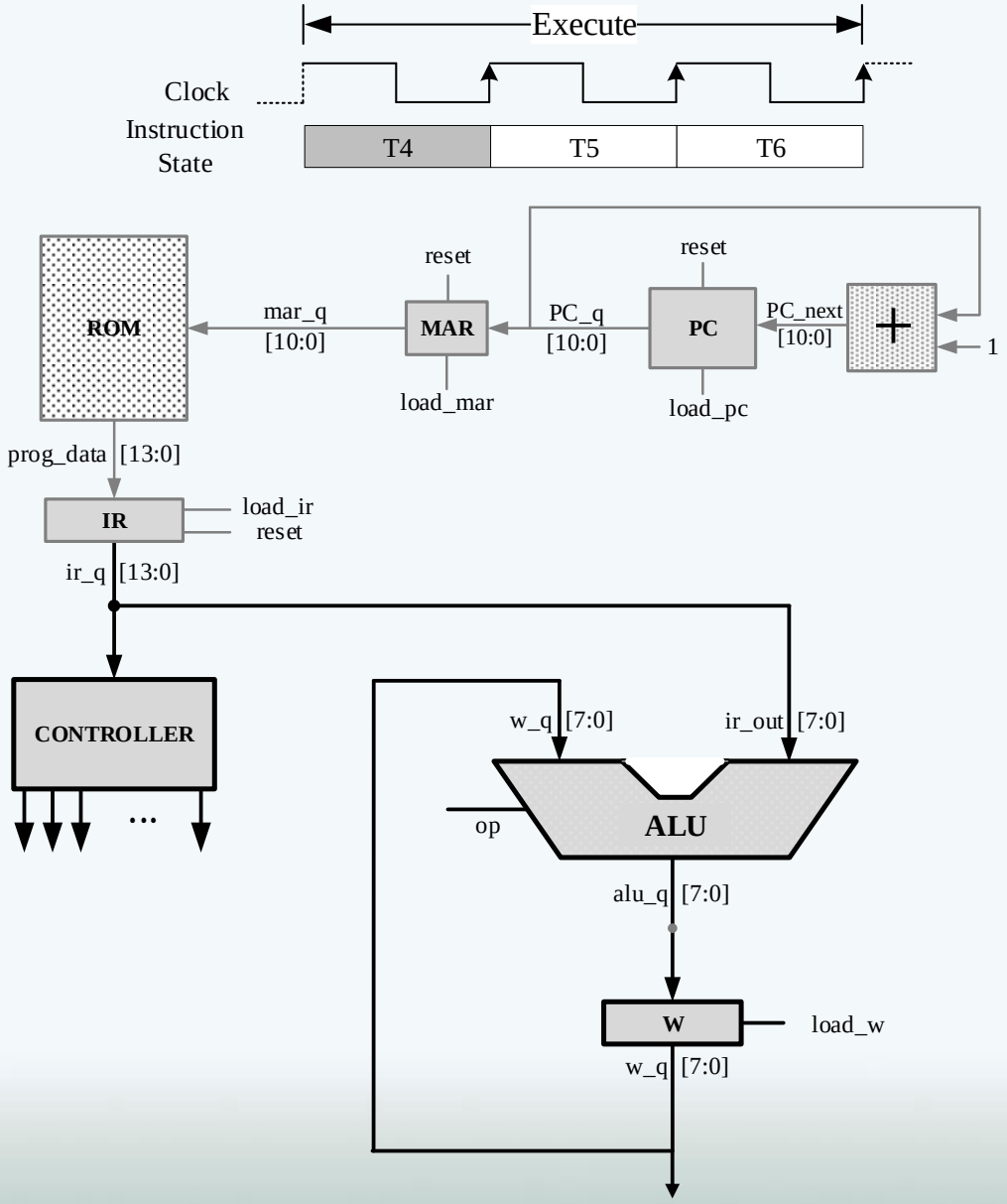
11 0000 kkkk kkkk



狀態	動作	控制訊號
T_4	W←IR[7:0]	op=5 load_w
T_5	無動作	無
T_6	無動作	無

ADDLW：立即數和W相加後傳回W

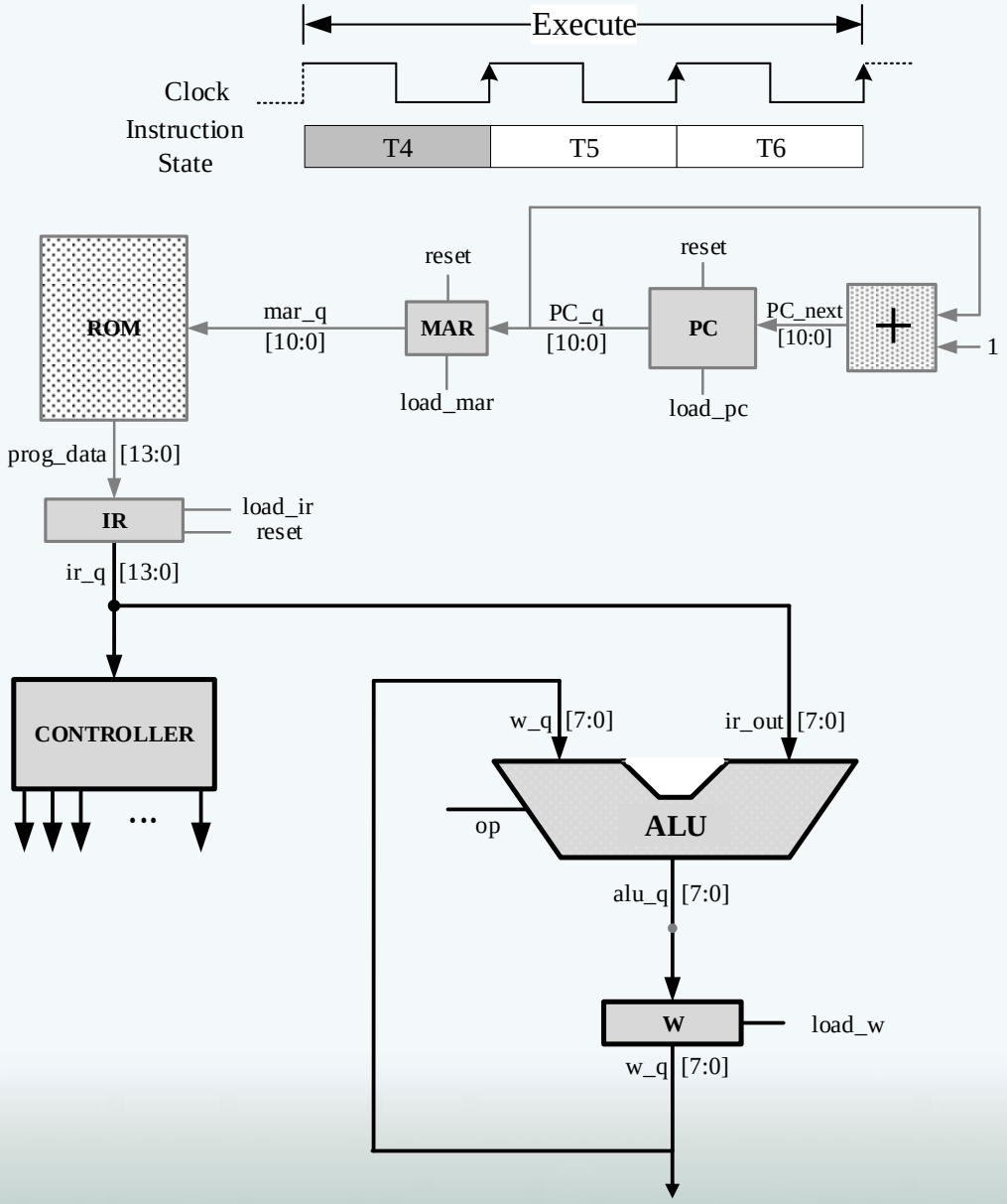
11 1110 kkkk kkkk



狀態	動作	控制訊號
T_4	W←IR[7:0]+W	op=0 load_w
T_5	無動作	無
T_6	無動作	無

IORLW：立即數和W作邏輯或運算後傳回W

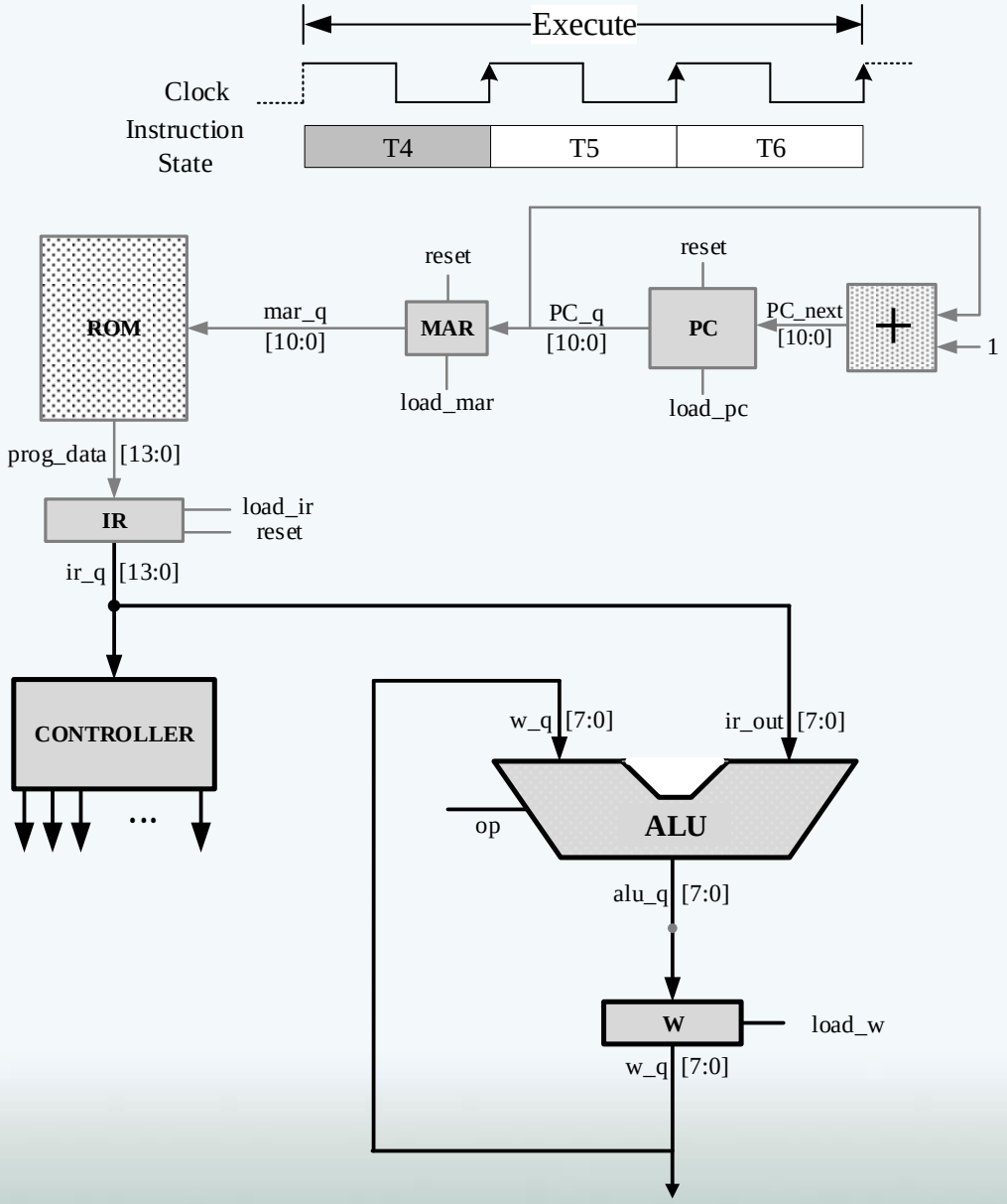
11 1000 kkkk kkkk



狀態	動作	控制訊號
T_4	W←IR[7:0] | W	op=3 load_w
T_5	無動作	無
T_6	無動作	無

SUBLW：立即數減去W的內容後傳回W

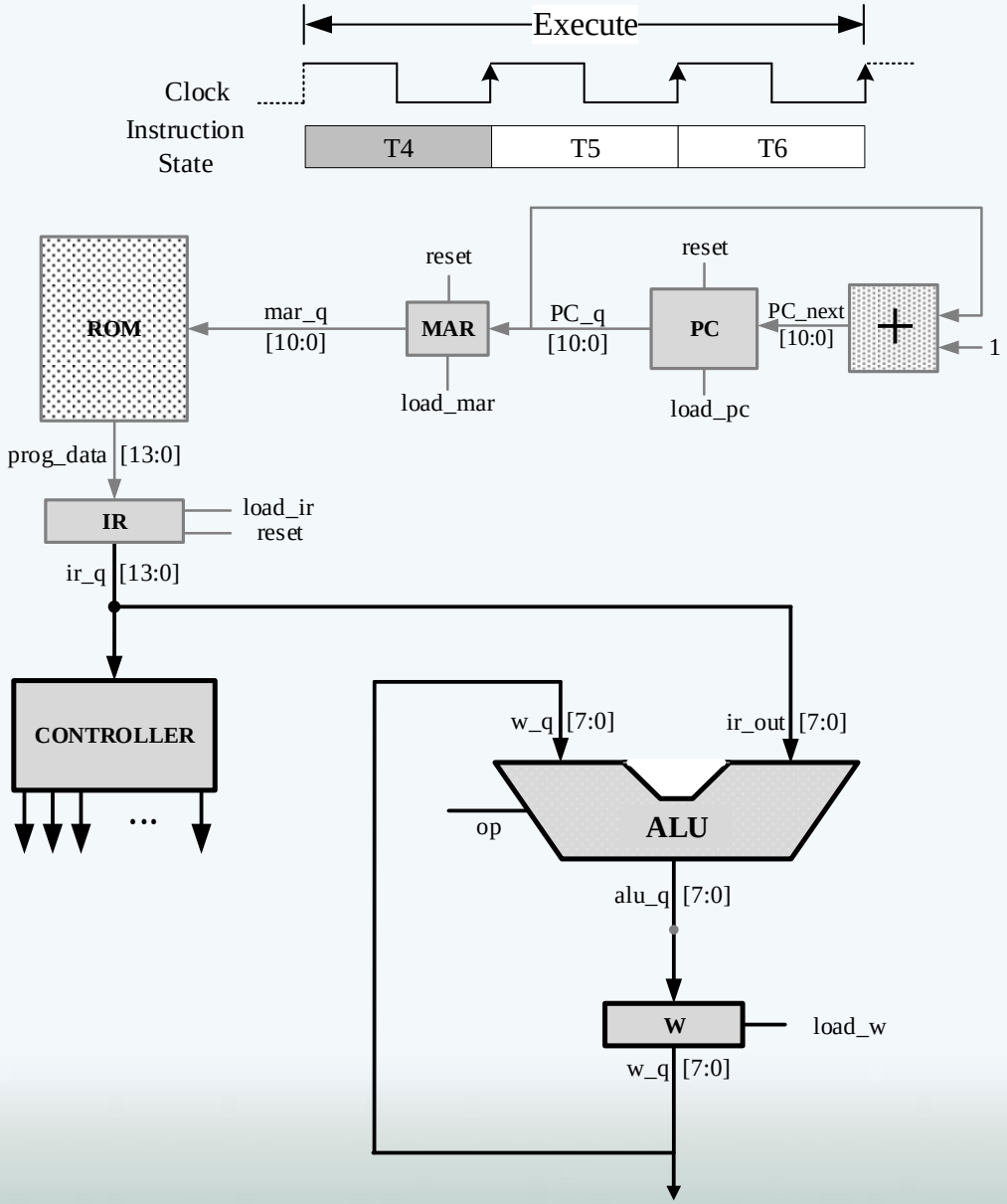
11 1100 kkkk kkkk



狀態	動作	控制訊號
T_4	W←IR[7:0]-W	op=1 load_w
T_5	無動作	無
T_6	無動作	無

ANDLW：立即數和W作邏輯與運算後傳回W

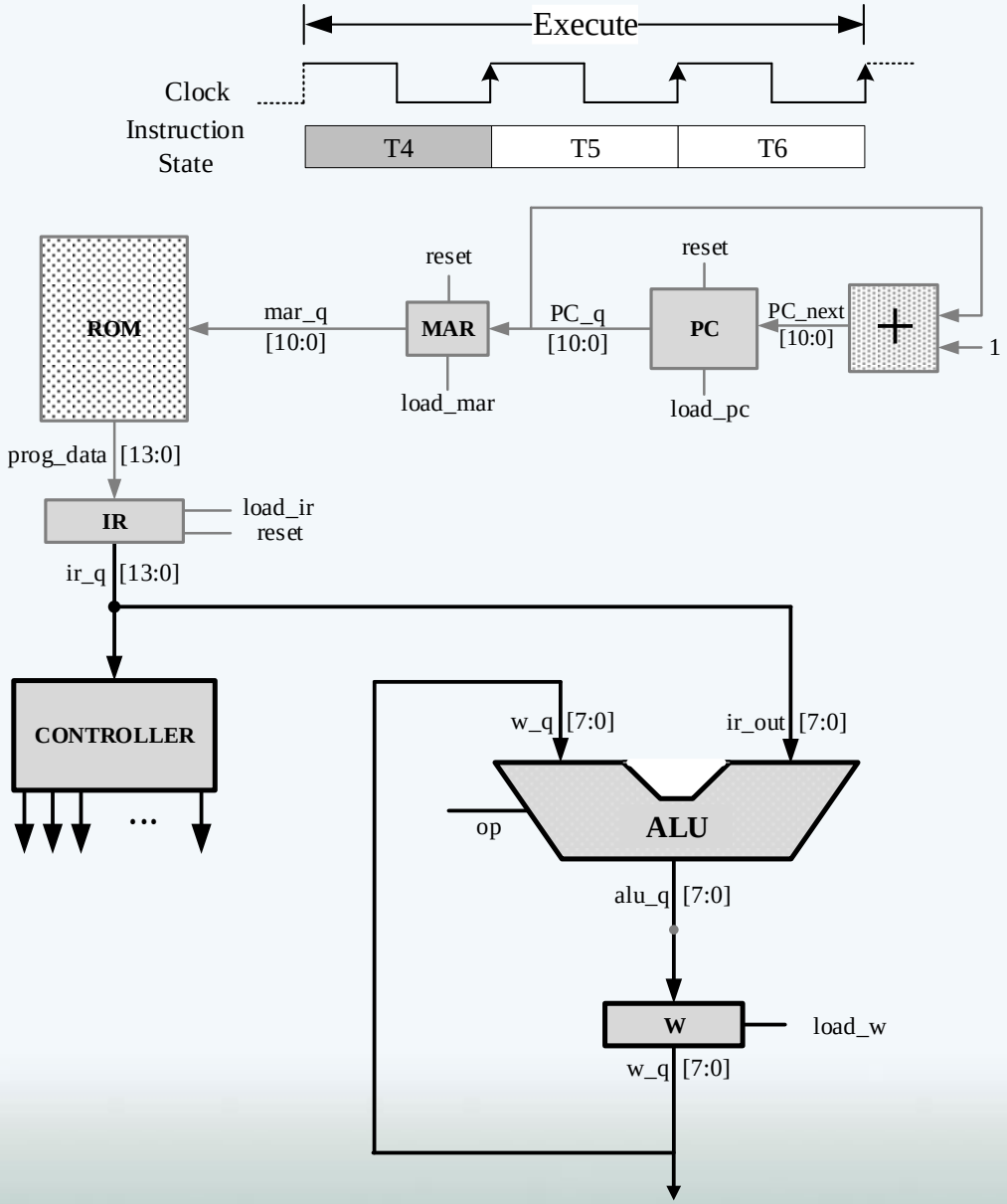
11 1001 kkkk kkkk



狀態	動作	控制訊號
T_4	W←IR[7:0] & W	op=2 load_w
T_5	無動作	無
T_6	無動作	無

XORLW：立即數和W作邏輯異或運算後傳回W

11 1010 kkkk kkkk



狀態	動作	控制訊號
T_4	W←IR[7:0] ^ W	op=4 load_w
T_5	無動作	無
T_6	無動作	無

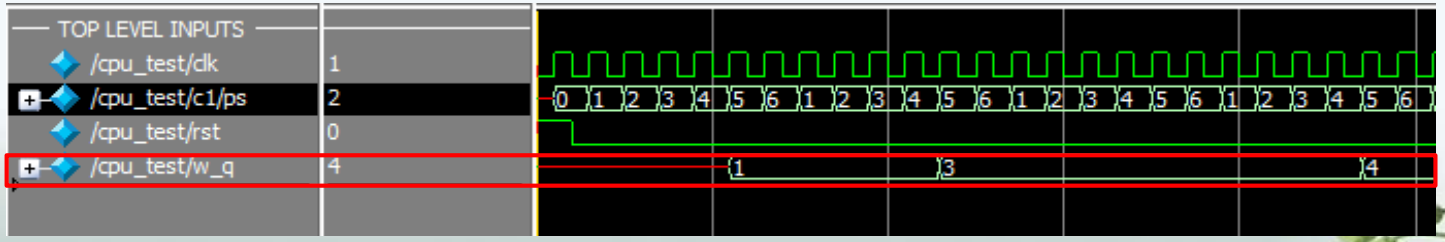
上課實作

```
module Program_Rom(
    output logic [13:0] Rom_data_out,
    input [10:0] Rom_addr_in
);
//---------

    logic [13:0] data;
    always_comb
    begin
        case (Rom_addr_in)
            11'h0:  data = 14'h3001;          //MOVLW 1
            11'h1:  data = 14'h3E02;          //ADDLW 2
            11'h2:  data = 14'h3003;          //MOVLW 3
            11'h3:  data = 14'h3004;          //MOVLW 4
            default:data = 14'h0;
        endcase
    end
     assign Rom_data_out = data;

endmodule
```

Program_Rom說明

MOVLW 1
位置0：W <= 1
ADDLW 2
位置1：W <= W + 2

MOVLW 3
位置2：W <= 3
MOVLW 4
位置3：W <= 4



作業

```
module Program_Rom(
    output logic [13:0] Rom_data_out,
    input [10:0] Rom_addr_in
);
//---------

    logic [13:0] data;
    always_comb
    begin
        case (Rom_addr_in)
            11'h0:  data = 14'h3044;        //MOVLW
            11'h1:  data = 14'h3E01;        //ADDLW
            11'h2:  data = 14'h3802;        //IORLW
            11'h3:  data = 14'h39FE;        //ANDLW
            11'h4:  data = 14'h3C47;        //SUBLW
            11'h5:  data = 14'h3A55;        //XORLW
            11'h6:  data = 14'h3AAA;        //XORLW
            default:data = 14'h0;
        endcase
    end
    assign Rom_data_out = data;

endmodule
```

Program_Rom說明

MOVLW 44
位置0：W <= 44

ADDLW 01
位置1：W <= W + 01

IORLW 02
位置2：W <= W | 02

ANDLW FE
位置3：W <= W & FE

SUBLW 47
位置4：W <= 47 - W

XORLW 55
位置5：W <= W ^ 55

XORLW AA
位置6：W <= W ^ AA