Microprocessor Interfacing & Programming

LECTURE 11&12

Delay Calculation for PIC18

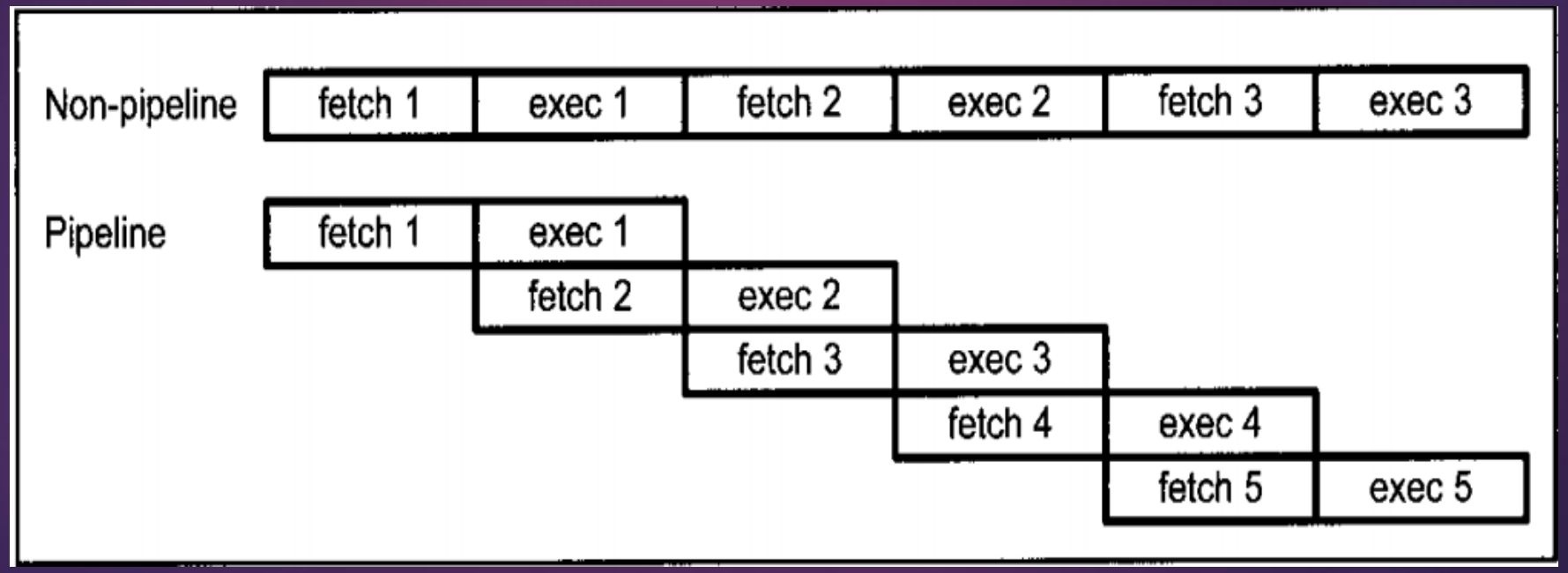
- 2 factors that can affect the accuracy of the delay:

1. Crystal Fequency: The frequency of the crystal oscillator connected to the OSC1 and OSC2 input pins is one factor in the time delay calculation. The duration of the clock period for the instruction cycle is a function of this crystal frequency.
2. PIC Design: PIC execute an instruction in 1 cycle. There are 3 ways to do that:

a. Harvard architecture.
b. RISC architecture.
c. Pipelining

Pipelining

- The concept of pipelining allow the CPU to fetch and execute the instructions at the same time.



Instruction cycle time for PIC

- It takes a certain amount of time for the processor to execute an instruction. In PIC, this time is referred as **instruction cycles**.
- As all the instructions in PIC18 are either 2 byte or 4 byte, most instructions take no more than one or two instruction cycles to execute.
- In PIC family, the length of the instruction cycle depends on the frequency of the oscillator connected to the PIC system.
- In PIC18, one instruction cycle is equal to **four times** the crystal oscillator clock period.
- Therefore, to calculate the instruction cycle for PIC, we take 1/4 of the crystal frequency.

- So, one instruction cycle in PIC18 is the basic unit of time in which the CPU fetches, decodes, and executes an instruction.
- For PIC18, 4 oscillator periods = 1 instruction cycle.

- The crystal (or internal oscillator) generates a clock signal at frequency F.
- This signal is very fast, but the CPU cannot directly execute instructions at that raw rate.

- The PIC18 uses a 4-step process for each instruction cycle:

a. Fetch the instruction from program memory

b. Decode the instruction

c. Execute the instruction (ALU operation, data move, etc.)

d. Write back or complete the cycle

- Each of these phases requires one oscillator tick.

Example

The following shows the crystal frequency for three different PIC-based systems. Find the period of the instruction cycle in each case.

(a) 4 MHz (b) 16 MHz (c) 20 MHz

Solution:

(a) 4/4 = 1 MHz; instruction cycle is 1/1 MHz = 1 μs (microsecond)

(b) 16 MHz/4 = 4 MHz; instruction cycle = 1/4 MHz = 0.25 μs = 250 ns (nanosecond)

(c) 20 MHz/4 = 5 MHz; instruction cycle = 1/5 MHz = 0.2 μs = 200 ns

Branch penalty

- For pipelining, we need a buffer or queue in which an instruction is prefetched and ready to be executed.
- In some cases, processor must flush out the queue. e.g; when a branch instruction is executed, the processor starts to fetch codes from the new memory location and the code in the queue that was fetched previously is discarded.
- In this case, the execution unit must wait until the fetch unit fetches the new instruction, called **branch penalty**.
- The branch penalty is an extra instruction cycle to fetch the instruction from the target location instead of executing the instruction right below the branch.

Contd...

- Due to this branch penalty, some instructions in PIC18 takes two or three instruction cycles.
- These are GOTO, BRA, CALL and all the conditional branch instructions such as BNZ, BC and so on.
- The conditional branch instruction can take only instruction cycle if it does not jump.

For a PIC18 system of 4 MHz, find how long it takes to execute each of the following instructions:

(a) MOVLW	(b) DECF	(c) MOVWF
(d) ADDLW	(e) NOP	(f) GOTO
(g) CALL	(h) BNZ	

Solution:

The machine cycle for a system of 4 MHz is 1 µs, as shown in Example 3-14. Appendix A shows instruction cycles for each of the above instructions. Therefore, we have:

Instruction	*Instruction cycles*	*Time to execute*
(a) MOVLW 0x55	1	1 × 1 µs = 1 µs
(b) DECF MYREG	1	1 × 1 µs = 1 µs
(c) MOVWF	1	1 × 1 µs = 1 µs
(d) ADDLW	1	1 × 1 µs = 1 µs
(e) NOP	1	1 × 1 µs = 1 µs
(f) GOTO	2	2 × 1 µs = 2 µs
(g) CALL	2	2 × 1 µs = 2 µs
(h) BNZ	2/1	(2 µs taken, 1 µs if it falls through)

Size of Delay Calculation

- Size of a delay = Time period of single instruction cycle (T) x Total no. of instruction cycles (N)

Find the size of the delay of the code snippet below if the crystal frequency is 4 MHz:

Solution:

From Appendix A, we have the following machine cycles for each instruction of the DELAY subroutine:

```
                                        Instruction Cycle
MYREG EQU   0x08          ;use location 08 as counter

DELAY     MOVLW     0xFF                  1
          MOVWF     MYREG                 1

AGAIN     NOP                             1
          NOP                             1
          DECF      MYREG,F               1
          BNZ       AGAIN                 2

          RETURN                          1
```

Therefore, we have a time delay of $[(255 \times 5) + 1 + 1 + 1] \times 1\ \mu s = 1278\ \mu s$.
Notice that BNZ takes two instruction cycles if it jumps back, and takes only one when falling through the loop. That means the above number should be 1277 μs.

Find the size of the delay in the following program if the crystal frequency is 4 MHz:

```
MYREG EQU    0x08                    ;use location 08 as counter

          ORG        0
BACK      MOVLW      0x55            ;load WREG with 55H
          MOVWF      PORTB           ;send 55H to port B
          CALL       DELAY           ;time delay
          MOVLW      0xAA            ;load WREG with AA (in hex)
          MOVWF      PORTB           ;send AAH to port B
          CALL       DELAY
          GOTO       BACK            ;keep doing this indefinitely

;————————  this is the delay subroutine
          ORG        300H            ;put time delay at address 300H
DELAY     MOVLW      0xFA            ;WREG = 250, the counter
          MOVWF      MYREG
AGAIN     NOP                        ;no operation wastes clock cycles
          NOP
          NOP
          DECF       MYREG, F
          BNZ        AGAIN           ;repeat until MYREG becomes 0
          RETURN                     ;return to caller
          END                        ;end of asm file
```

Solution:

From Appendix A, we have the following machine cycles for each instruction of the DELAY subroutine:

			Instruction Cycle
DELAY	MOVLW	0xFA	1
	MOVWF	MYREG	1
AGAIN	NOP		1
	NOP		1
	NOP		1
	DECF	MYREG, F	1
	BNZ	AGAIN	2
	RETURN		1

Therefore, we have a time delay of $[(250 \times 6) + 1 + 1 + 1] \times 1\ \mu s = 1503\ \mu s$.

For a instruction cycle of 1 μs, find the time delay in the following subroutine:

```
R2      EQU     0x7
R3      EQU     0x8
DELAY                           Instruction Cycle
                MOVLW   D'200'      1
                MOVWF   R2          1
AGAIN           MOVLW   D'250'      1
                MOVWF   R3          1
HERE            NOP                 1
                NOP                 1
                DECF    R3, F       1
                BNZ     HERE        2
                DECF    R2, F       1
                BNZ     AGAIN       2
                RETURN              1
```

Loop inside a loop delay

For the HERE loop, we have (5 × 250) 1 μs = 1250 μs. The AGAIN loop repeats the HERE loop 200 times; therefore, we have 200 × 1250 μs = 250000 μs, if we do not include the overhead. However, the following instructions of the outer loop add to the delay:

```
AGAIN       MOVLW       D'250'      1
            MOVWF       R3          1
            .....
            DECF        R2, F       1
            BNZ         AGAIN       2
```

The above instructions at the beginning and end of the AGAIN loop add 5 × 200 × 1 μs = 1000 μs to the time delay. We should also subtract 200 μs for the times BNZ HERE falls through. As a result we have 250000 + 1000 – 200 = 250800 μs = 250.8 milliseconds for the total time delay associated with the above DELAY subroutine. Notice that in the case of a nested loop, as in all other time delay loops, the time is approximate because we have ignored the first few instructions and the last instruction, RETURN, in the subroutine. NOP is a 2-byte instruction. There are 11 instructions in the above DELAY program, and all the instructions are 2-byte instructions. That means that the loop delay takes 22 bytes of ROM code space.

Example 3-19

Find the time delay for the following subroutine, assuming a crystal frequency of 4 MHz. Discuss the disadvantage of this over Example 3-18.

```
MYREG EQU   0x8
                                    Machine Cycle

DELAY     MOVLW     D'200'              1
          MOVWF     MYREG               1

AGAIN     NOP                           1
          NOP                           1
          NOP                           1
          NOP                           1
          NOP                           1
          NOP                           1
          NOP                           1
          NOP                           1
          NOP                           1
          NOP                           1
          NOP                           1
          NOP                           1
          NOP                           1
          DECF      MYREG,  F           1
          BNZ       AGAIN               2

          RETURN                        1
```

Solution:

The time delay inside the AGAIN loop is $[200(13 + 2)] \times 1\ \mu s = 3000\ \mu s$. NOP is a 2-byte instruction, even though it does not do anything except to waste cycle time. There are 17 instructions in the above DELAY program, and all the instructions are 2-byte instructions. This means the loop delay takes 34 bytes of ROM code space, and gives us only a 3000 μs delay. That is the reason we use a nested loop instead of NOP instructions to create a time delay. Chapter 9 shows how to use PIC timers to create delays much more efficiently.

Write a program to toggle all the bits of SFR PORTB every 1 s. Assume that the crystal frequency is 10 MHz and the system is using a PIC18F458.

Solution:

```
;tested using MPLAB with PIC18F458 operating at 10 MHz
R2      EQU     0x2
R3      EQU     0x3
R4      EQU     0x4


            MOVLW       0x55            ;load WREG with 55H
            MOVWF       PORTB           ;send 55H to PORTB B
BACK        CALL        DELAY_500MS     ;time delay
            COMF        PORTB           ;complement PORTB
            GOTO        BACK            ;keep doing this indefinitely

;--------   this is the delay subroutine
DELAY_500MSEC
                MOVLW           D'20'
                MOVWF           R4
BACK            MOVLW           D'100'
                MOVWF           R3
AGAIN           MOVLW           D'250'
                MOVWF           R2
HERE            NOP
                NOP
                DECF            R2, F
                BNZ             HERE
                DECF            R3, F
                BNZ             AGAIN
                DECF            R4, F
                BNZ             BACK
                RETURN
```

Delay $20 \times 100 \times 250 \times 5 \times 400$ ns $= 1{,}000{,}000{,}000$ ns $= 1{,}000{,}000$ µs $= 1$ s.

Calculation for counters practice

- Do the calculations for the counter values if oscillator frequency is 4mHz and the required delay is 2 seconds.

- The required delay is divided by the time period of a single instruction cycle (T).

PIC multistage execution pipeline

- In PIC18, the execution unit takes 4 clock periods of the oscillator.
- That is why, we divide the oscillator by 4 to get the instruction cycle.

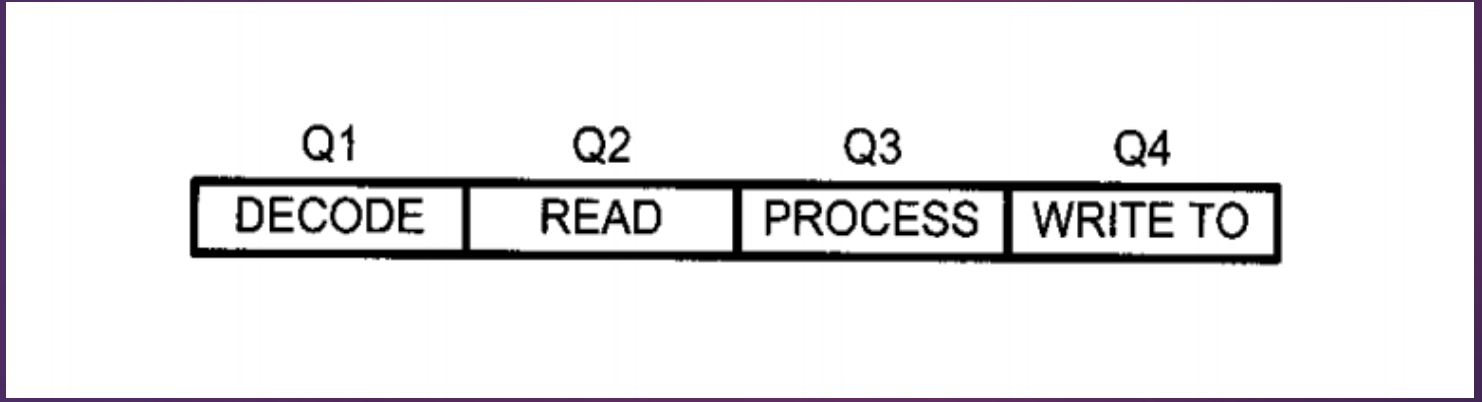
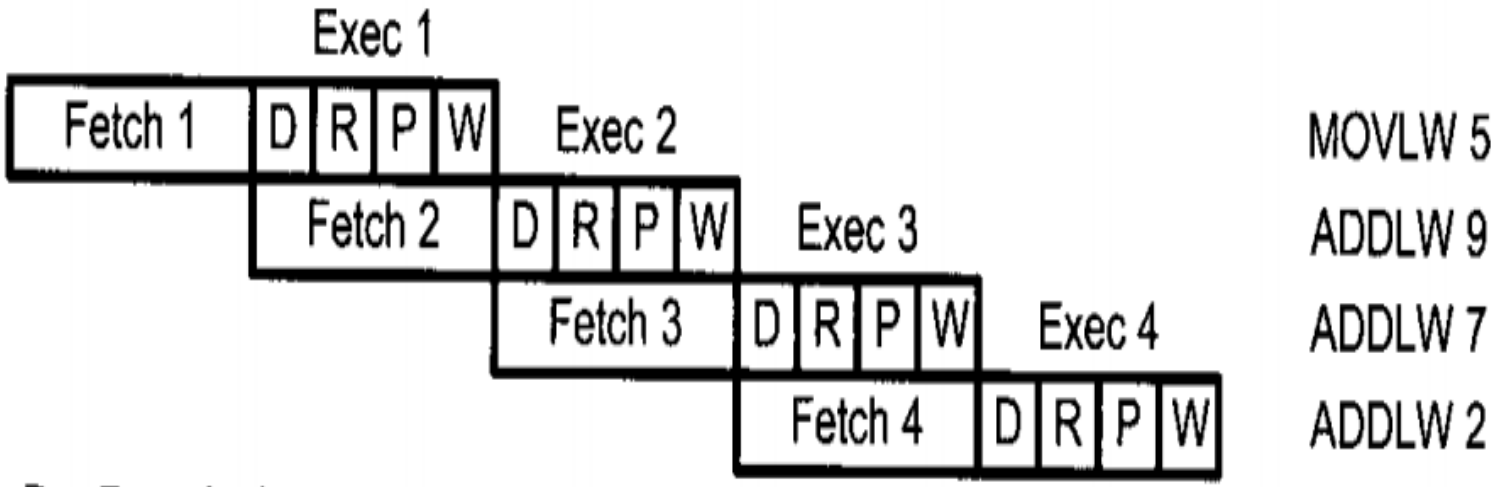


Figure 3-10 explains why we divide the oscillator by 4 to get the instruction cycle. In Q1, we decode the instruction that is already fetched and sitting in the queue. In Q2, the operand is fetched from the file register. In Q3, the operation is performed: The adding of the two numbers is done. In Q4, the result is written into the destination register. In reality, one can construct the PIC18 superpipeline for four instructions, and is shown in Figure 3-11.



D = Decode the instruction

R = Read the operand

P = Process (eg. ADDLW)

W = Write the result to destination register

Find the oscillator frequency if the instruction cycle = 1.25 μs.

Find the instruction cycle if the crystal frequency is 20 MHz.

Find the instruction cycle if the crystal frequency is 10 MHz.

Find the instruction cycle if the crystal frequency is 16 MHz.

Find the time delay for the delay subroutine shown below if the system has a PIC18 with a frequency of 4 MHz:

```
        MOVLW D'200'
        MOVWF REGA
BACK    MOVLW D'250'
        MOVWF REGB
HERE    NOP
        DECF    REGB
        BNZ     HERE
        DECF    REGA
        BNZ     BACK
```