EE461 Microprocessor-based Digital Design

Stepper Motors

Assignment 16 Solutions

1. A certain stepper motor has 400 steps per revolution. To achieve finer control of its motion, the PIC16F874 controlling it operates the motor in half-step mode. In doing this, the least significant four bits of PORTB are allocated as digital outputs driving terminals *A, B, C,* and *D* of a six-wire stepper motor. The partially documented program in Listing 1 on page 4 repeatedly cycles through the eight possible control patterns and places them in PORTB. Assume that the PIC16F874 has a crystal oscillator with frequency f_{OSC} = 10 MHz.

   (a) With this program how much time elapses between the issuance of each successive stepper-motor control pattern?

   SOLUTION

   The control loop extends from line 26 to line 34, inclusive. However, it includes a `call` to subroutine `GetStep` in line 27. All instructions take one program cycle, unless they modify the program counter. In each cycle through the loop, the `call` instruction in line 27, the `goto` instruction in line 34, and the `addwf` instruction in line 38 modify the program counter. So does whichever one of the eight `ret lw` instructions in lines 39–46 executes. This makes four instructions requiring two cycles apiece, or eight cycles all told. In addition, there are four instructions that do not modify the program counter and so require only one cycle apiece, making for four additional cycles. This makes a grand total of 12 instruction cycles to complete the loop each time. The total time T required for each output pattern, then, is

$$
\begin{aligned}
T &= 12T_{INST} \\
&= \frac{12 \times 4}{f_{OSC}} \\
&= \frac{12 \times 4}{10\,\text{MHz}} \\
&= 4.8\ \mu\text{s}.
\end{aligned}
$$

(b) Assuming the motor can keep up with the control signals, what is the rotational velocity of the stepper motor? Give your answer to the nearest rpm.

SOLUTION

It takes 400 full steps or 800 half steps to complete a revolution. Therefore, the rotational period T_{ROT} is given by

$$T_{ROT} = 800T$$
$$= 800 \times 4.8\ \mu s$$
$$= 3.840 ms.$$

The rotational velocity f is

$$f = 1/T_{ROT}$$
$$= 1/3.840 ms$$
$$\mu\ \underline{60s}\ ¶$$
$$= 260.4 Hz 1\ min$$
$$= 15625\ rpm.$$

(c) Describe the changes you would need to make to the program to cause the motor to change directions. If this changes the motor's speed, what will be the new speed?

SOLUTION

One way to change the direction would be to reverse the sequence of `retlw` instructions in the `GetStep` subroutine. However, an easier way to achieve reversal would be to replace the `incf` instruction in line 31 with a `decf` instruction. However, when variable `i` is decremented from 0, the eight-bit result is $1111\ 1111_2$. We want the pattern to be 00000111_2. We could get this by performing an `and` operation on the value of `i` using the mask 00000111.

These changes would require replacing the single `bcf` instruction in line 33 with a `movlw` instruction and an `andwf` instruction, lengthening the entire loop by one instruction cycle. The result would be a lowered rotational velocity of 14423 rpm.

(d) Describe how you would modify the program to achieve full-step control. What rotational velocity would result now, again assuming the motor can keep up with the control signals? Give your answer to the nearest rpm.

SOLUTION

One way to achieve full-step control would be to remove all output patterns that entail activating two output pins at a time, namely,

the `retlw` instructions with operands `StepAB`, `StepBC`, `StepCD`, or `StepDA`. This would require changing the `bcf` instruction in line 33 so that it clears bit 2 instead of bit 3, since `i` now only needs to count from 0 to 3 and so needs only two bits, not three. The effect would be to double the rotational velocity to 31 250 rpm.

An alternative would be to add a second `incf i , F` instruction between lines 31 and 32. This would increment `i` by 2, rather than 1, each time, effectively skipping the output patterns that activate two stepper-motor coils at a time.

Another solution would be to replace the `incf` instruction with the two-instruction sequence

```
movlw   2
addwf   i,F
```

which does the same thing.

These last two options would alter the rotational velocity in two ways. Going to full-step control would increase it, but adding an extra instruction in the loop would serve to decrease it. As seen above, adding one single-cycle extra instruction to the control loop reduces the half-step rotational velocity to 14423 rpm. Going to full-step control would double this, yielding 28846 rpm.

Listing 1: A Stepper Motor Control Program

```
1                #include <p16f874.inc>
2 ; Four LSB of Port B are outputs to a stepper motor.
3 TRISBInit equ        B'1111OOOO'
4 ; Stepper motor output patterns
5 StepA      equ       B'OOOO1OOO'
6 StepAB     equ       B'OOO011OO'
7 StepB      equ       B'OOOOO1OO'
8 StepBC     equ       B'OOOO011O'
9 StepC      equ       B'OOOOOO1O'
10 StepCD    equ       B'OOOOO011'
11 StepD     equ       B'OOOOOOO1'
12 StepDA    equ       B'OOOO1OO1'
13 i         equ       H'22'      ; Storage for counter i
14           org       0          ; Reset vector
15           movlw     StepA      ; Initialize Port B to Step A
16           movwf     PORTB
17           bsf       STATUS,RPO       ; Bank1
18           movlw     TRISBInit        ; Initialize Port B
             bit       direction
19           movwf     TRISB
20           bcf       STATUS,RPO       ; Bank 0
21           clrf      i                ; Initialize i=0
22 ; while{TRUE} {
23 Next Step
24           movf      i,W              ; Find output pattern
25           call      GetStep          ; for step i
26           movwf     PORTB
27 ; Advance to next step
28           incf      i,F
29 ; Only 3 least significant bits       of i matter
30           bcf       i,3
31           goto      NextStep
32 ;}
33 ; Retrieve the requested stepper-motor pattern
34 GetStep addwf       PCL,F
35           retlw     StepA
36           retlw     StepAB
37           retlw     StepB
38           retlw     StepBC
39           retlw     StepC
40           retlw     StepCD
41           retlw     StepD
42           retlw     StepDA
43           end
```