VLF Comb Generator

E. Paschal, Hayden Gebhardt

January 10, 2024

Contents

1 Background

The programs below simulate an 11-stage shift register with feedback to produce a maximal-length shift register sequence. Such a sequence is a pseudo-random sequence. Its auto-correlation is 2047 for offsets of 0 and multiples of 2047, and -1 for all other offsets. The spectrum of the output of the device is a comb spectrum.

With an oscillator input of 16.384 MHz, the internal clock runs at 16.384/4 or 4.096 MHz. The cycle time for the shift register loop is 8 instructions, so the virtual clock for the shift register is 512 kHz. With a cycle length of 2047 shifts, the pseudo-random sequence repeats at a rate of 250.1221 Hz, which is the spacing of the components of the comb spectrum.

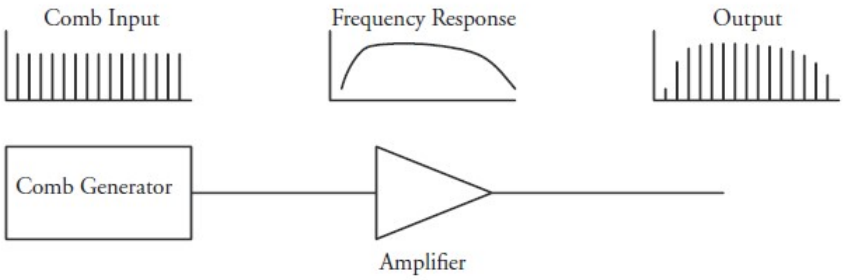

Figure 1: Illustration of expected responses for comb calibration.

The microprocessor used to execute this code is the PIC12F629. Its pin configuration can be seen here:

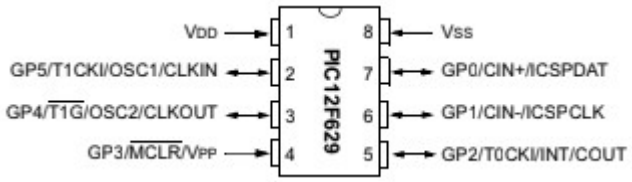

Figure 2: PIC12F629 pin-out

2 Annotated Code

Listing 1: Initial Configuration

```
; Processor Inclusion
;*************************************************************

    LIST p=12F629          ;processor
    include <p12f629.inc> ; header file for MC.

; Configuration Word
;*************************************************************

; CONFIG
    __CONFIG _BODEN_ON   ; Brown out detection enabled
    __CONFIG _MCLRE_OFF ; Master clear unset, GP3 is GPIO
    __CONFIG _PWRTE_OFF ; Power-up timer disabled
    __CONFIG _WDT_OFF    ; Watchdog timer disabled
    __CONFIG _EC_OSC     ; Oscillator set to external clock
                         ; External clock is 16 MHz

;Variable definitions
;*************************************************************

plusout  EQU b'00010100' ; +CAL high, -CAL low
feedback EQU 0xEB        ; Feedback bits for 2047-long
                         ; sequence, this particular
                         ; feedback also sets +CAL low,
                         ; -CAL high as GPIO
byteL    EQU 0x20        ; Lower bits allocated for LFSR
byteH    EQU 0x21        ; Upper bits allocated for LFSR
C        EQU b'00000000' ; Carry register starting seed
```

Listing 2: Start sequence, GPIO configuration, and beginning of loop

```
; Reset Vector
;*****************************************************************
    ORG     0x0
    GOTO    Start ; Go to beginning of program

; Driver Core
;*****************************************************************
    ORG     0x010
Start
    BANKSEL TRISIO      ; Set bank that contains TRISIO
    MOVLW   b'11101000'
    MOVWF   TRISIO      ; Set GPIO 0,1,2,4 as outputs

    BANKSEL CMCON       ; Set bank that contains CMCON
    MOVLW   b'00000111'
    MOVWF   CMCON       ; Set GPIO 0,1,2 to digital I/O
    CLRF    byteL       ; Clear registers to initialize
    CLRF    byteH
    Loop0               ; Start of infinite loop
    MOVLW   plusout
    MOVWF   GPIO        ; Carry 1 - high output
    BCF     STATUS, C
    RLF     byteL, F
    RLF     byteH, F    ; Shift registers left
    BTFSC   byteH, 3    ; Test bit 12 (carry out)
    GOTO    Loop0       ; Carry 1
    MOVLW   feedback    ; 1st carry 0
    XORWF   byteL, F    ; Insert feedback
    MOVWF   GPIO        ; Low output
    BCF     STATUS, C
    RLF     byteL, F
    RLF     byteH, F    ; Shift registers left
    BTFSC   byteH, 3    ; Test bit 12 (carry out)
```

Listing 3: Full loop instructions and overflow catch

```
    GOTO    Loop0           ; Carry = 1
    MOVLW   feedback        ; 2nd carry = 0
    XORWF   byteL, F        ; Insert feedback
    MOVWF   GPIO            ; Low output
    BCF     STATUS, C
    RLF     byteL, F
    RLF     byteH, F        ; Shift registers left
    BTFSC   byteH, 3        ; Retest bit 12 (carry out)

    ; ... [Instructions for carries 3-8] ...

    GOTO    Loop0           ; Carry = 1
    MOVLW   feedback        ; 9th carry = 0
    XORWF   byteL, F        ; Insert feedback
    MOVWF   GPIO            ; Low output
    BCF     STATUS, C
    RLF     byteL, F
    RLF     byteH, F        ; Shift registers left
    BTFSC   byteH, 3        ; Retest bit 12 (carry out)
    GOTO    Loop0           ; Carry = 1
    MOVLW   feedback        ; 10th carry = 0
    XORWF   byteL, F        ; Insert feedback
    MOVWF   GPIO            ; Low output
    BCF     STATUS, C
    RLF     byteL, F
    RLF     byteH, F        ; Shift registers left
    BTFSC   byteH, 3        ; Retest bit 12 (carry out)
    GOTO    Loop0           ; Carry = 1

Loop1
    MOVLW   feedback        ; 11th carry 0 (cannot happen)
    MOVWF   GPIO            ; Sequence cannot exceed
                            ; maximal length, set pins low
    GOTO    Loop1           ; Repeat until restart

    END     ;End of program
```

3 Appendix

3.1 Assembly Notes

This code was written in MPLABX ver. 5.35 using the MPASM compiler. The configuration word and pre-processor instructions are defined specifically for the PIC12F629 microcontroller. The number of loop instructions correspond to an 11-stage linear feedback shift register, designed to produce spectral combs spaced 244.26 Hz apart.

3.1.1 Source Code

```
; Processor Inclusion
;*******************************************************

    LIST p=12F629    ;processor
    #include <p12f629.inc>

; Configuration Word
;*******************************************************

; CONFIG
; __config 0x31F3
    __CONFIG _BOREN_ON & _MCLRE_OFF & _PWRTE_OFF & _WDT_OFF &
    _EC_OSC

;Variable definitions
;*******************************************************

plusout     EQU     b'00010100'
feedback    EQU     0xEB
byteL       EQU     0x20
byteH       EQU     0x21
C       EQU     b'00000000'

; Reset Vector
;*******************************************************

    ORG     0x0
    GOTO    Start

; Driver Core
;*******************************************************
```

```
31
32      ORG     0x010
33  Start
34      BANKSEL TRISIO
35      MOVLW   b'11101000'
36      MOVWF   TRISIO
37
38      BANKSEL CMCON
39      MOVLW   b'00000111'
40      MOVWF   CMCON
41      CLRF    byteL
42      CLRF    byteH
43
44  Loop0
45      MOVLW   plusout
46      MOVWF   GPIO
47      BCF     STATUS, C
48      RLF     byteL, F
49      RLF     byteH, F
50      BTFSC   byteH, 3
51      GOTO    Loop0
52      MOVLW   feedback
53      XORWF   byteL, F
54      MOVWF   GPIO
55      BCF     STATUS, C
56      RLF     byteL, F
57      RLF     byteH, F
58      BTFSC   byteH, 3
59      GOTO    Loop0
60      MOVLW   feedback
61      XORWF   byteL, F
62      MOVWF   GPIO
63      BCF     STATUS, C
64      RLF     byteL, F
65      RLF     byteH, F
66      BTFSC   byteH, 3
67      GOTO    Loop0
68      MOVLW   feedback
69      XORWF   byteL, F
70      MOVWF   GPIO
71      BCF     STATUS, C
72      RLF     byteL, F
73      RLF     byteH, F
74      BTFSC   byteH, 3
75      GOTO    Loop0
```

```
76      MOVLW   feedback
77      XORWF   byteL, F
78      MOVWF   GPIO
79      BCF     STATUS, C
80      RLF     byteL, F
81      RLF     byteH, F
82      BTFSC   byteH, 3
83      GOTO    Loop0
84      MOVLW   feedback
85      XORWF   byteL, F
86      MOVWF   GPIO
87      BCF     STATUS, C
88      RLF     byteL, F
89      RLF     byteH, F
90      BTFSC   byteH, 3
91      GOTO    Loop0
92      MOVLW   feedback
93      XORWF   byteL, F
94      MOVWF   GPIO
95      BCF     STATUS, C
96      RLF     byteL, F
97      RLF     byteH, F
98      BTFSC   byteH, 3
99      GOTO    Loop0
100     MOVLW   feedback
101     XORWF   byteL, F
102     MOVWF   GPIO
103     BCF     STATUS, C
104     RLF     byteL, F
105     RLF     byteH, F
106     BTFSC   byteH, 3
107     GOTO    Loop0
108     MOVLW   feedback
109     XORWF   byteL, F
110     MOVWF   GPIO
111     BCF     STATUS, C
112     RLF     byteL, F
113     RLF     byteH, F
114     BTFSC   byteH, 3
115     GOTO    Loop0
116     MOVLW   feedback
117     XORWF   byteL, F
118     MOVWF   GPIO
119     BCF     STATUS, C
120     RLF     byteL, F
```

```
121     RLF     byteH, F
122     BTFSC   byteH, 3
123     GOTO    Loop0
124     MOVLW   feedback
125     XORWF   byteL, F
126     MOVWF   GPIO
127     BCF     STATUS, C
128     RLF     byteL, F
129     RLF     byteH, F
130     BTFSC   byteH, 3
131     GOTO    Loop0
132
133 Loop1
134     MOVLW   feedback
135     MOVWF   GPIO
136     GOTO    Loop1
137
138     END
```

3.2 C Notes

This code will produce the same result as the assembly code above, but the XC8 compiler is not fast enough to generate in-phase outputs on all required GPIO pins. If one tethers two identical output pins, they would see a voltage division in the output due to the delay incurred by down-compiling the C code. Use this code if you have a faster micro-controller or a more reliable C compiler. This program was last executed on MPLAB ver. 6.10 using the supported XC8 compiler.

3.2.1 Source Code

```
1 #include <xc.h>
2 #include <stdio.h>
3 #include <stdint.h>
4 #include <stdlib.h>
5
6 // Configuration bits
7 #pragma config FOSC = EC       // GP4 operates as I/O, CLKIN on
      GP5
8 #pragma config WDTE = OFF     // Watchdog timer disabled
9 #pragma config PWRTE = OFF    // Power-up timer disabled
```

```
10 #pragma config MCLRE = OFF     // MCLR pin function is digital I/
   O
11 #pragma config BOREN = OFF     // Brown-out reset disabled
12 #pragma config CP = OFF        // Code protection disabled
13 #pragma config CPD = OFF       // Data code protection disabled
14
15 void main()
16 {
17     TRISIO &= 0xE8;  // set TRISIO (0, 1, 2, 4) as output
18     GPIO = 0;        // set all declared GPIO pins as outputs
19
20     const uint16_t start = 0x002; // algorithm seed (0b 0000
   0000 0010)
21     const uint16_t mask = 0x07FF; // constant mask for
   autocorrelation range (0d 2047))
22     uint16_t a = start;           // set autocorrelation equal
   to starting seed
23
24     while (1) {
25
26         unsigned int newbit = ((a >> 10) ^ (a >> 1)) & 1; //
   generate new bit using LSFR logic with two taps at 2^10, 2^1
   bits
27         a = ((a << 1) | newbit) & mask; // apply the new bit and
    mask the value to the autocorrelation range
28
29         // ternary operation to determine if sequence is
   complete. If so, set all GPIO outputs equal to seed
30         GPIO = a == start ? (start & 0xFF) : (a & 0xFF << 4) |
   (~a & 0x03); // set GPIO pins based on the autocorrelation
   value
31     }
32 }
```