GPS Signal Description

1. The baseband signal **transmited** by the satellite is given as

$$S(t) = S_{PPS}(t) + jS_{SPS}(t) \tag{1}$$

- $S_{SPS}(t) = \sum_{i=-\infty}^{\infty} c_{sps}(|i|_{L\_sps}).d([i]_{CD\_sps}).rect_{T_{c,sps}}(t - iT_{c,sps})$ — Standard Positioning Service
- $S_{PPS}(t) = \sum_{i=-\infty}^{\infty} c_{pps}(|i|_{L\_pps}).d([i]_{CD\_pps}).rect_{T_{c,pps}}(t - iT_{c,pps})$ —- Precision Positioning Service

2. Let $x_{in}[n]$ be the incoming signal at the **receiver** end and is given as

$$x_{in}[n] = A(t)s_T(t - \tau(t))e^{j(2\pi f_D(t)t+\phi(t))}|_{t=nT_s} + n(t)|_{t=nT_s} \tag{2}$$

where

- $A(t)$ is Amplitude
- $s_T(t)$ is Complex baseband signal
- $\tau(t)$ is code delay(time varying)
- $f_D(t)$ is Doppler shift(time varying)
- $\phi(t)$ is carrier phase shift(time varying)
- $n(t)$ is Random noise with zero mean
- T_s is Sampling period
- f_s is Sampling frequency

Pseudo code for GPS Signal Acquisition

1.1 Functions for computing the PRN codes of GPS satellite

1. (a) g1_lfsr()

```
state = 0x3FF
Declare an array out[1023]
for i=0 to i=1022
    out[i] = (state ≫ 9) & 0x1
    new_bit = ((state ≫ 9) ⊕ (state ≫ 2)) & 0x1
    state = ((state ≪ 1) | new_bit) & 0x3FF
end for
return out
```

(b) g2_lfsr(tap0,tap1)

```
state = 0x3FF
Declare an array out[1023]
tap0 = tap0-1
tap1 = tap1-1
for i=0 to i=1022
    out[i] = ((state ≫ tap0) ⊕ (state ≫ tap1)) & 0x1
    new_bit = ((state ≫ 9) ⊕ (state ≫ 8) ⊕ (state ≫ 7) ⊕ (state ≫ 5) ⊕ (state ≫ 2) ⊕ (state ≫ 1)) & 0x1
    state = ((state ≪ 1) | new_bit) & 0x3FF
end for
return out
```

(c) combine_g1_g2(g1,g2)

```
declare out[1023]
for i=0 to i=1022
    out[i] = g1[i] ⊕ g2[i]
end for
return out
```

1.2 Main function

1. PRN Code frequency f_c is 1.023Mhz
2. Sampling frequency f_s is 2.048Mhz
3. The number of samples **N** for 1ms is 2048
4. Static array SVs[64] = [2, 6, 3, 7, 4, 8, 5, 9, 1, 9, 2,10, 1, 8, 2, 9, 3,10, 2, 3, 3, 4, 5, 6, 6, 7, 7, 8, 8, 9, 9,10, 1, 4, 2, 5, 3, 6, 4, 7, 5, 8, 6, 9, 1, 3, 4, 6, 5, 7, 6, 8, 7, 9, 8,10, 1, 6, 2, 7, 3, 8, 4, 9] /* Array to store Phase selector values for 32 satellites - 2 values per satellite*/
5. Static array g1[1023]
6. static array g2[32][1023]

   /* Generating PRN code, BPSK modulation and upsampling for all 32 satellites */

7. g1 = g1_lfsr() /* Fucntion call */
8. k=0
9. visibile_satellites = 0
10. **for sv**=01 to **sv**=32:

    tap0 = SVs[k++]
    tap1 = SVs[k++]
    g2[**sv**] = g2_lfsr(tap0,tap1) /* Fucntion call */
    gold_code = combine_g1_g2(g1,g2) /* Fucntion call */

    /*BPSK modulation of PRN code */
    for i = 0 to **i** = 1022
      if gold_code[**i**] > 0
        gold_code[**i**] = -1
      else
        gold_code[**i**] = 1
    /* Upsampling the PRN code */
    for i = 0 to **i** = N-1
      c[**sv**][i] = gold_code[int(**i**× 0.49951171875)]
      C[**sv**][i] = conjugate of FFT(c[**sv**][i])

    end for

11. Capture 2ms samples of incoming signal $x_{in}[n]$
12. Calculate received signal power using the formula

    $P_x = 0$

    for i=0 to N-1

    $P_x = P_x + (x_{in}[n] \times x^*_{in}[n])$

13. The power of incoming signal should be $P_x>$ threshold . If true, **go to** step **14**. else, **stop** the process
14. Initialize the array max_power[5] = {0}
15. Initialize the array visible_satellites_withMaxPower[5] = {0}
16. Initialize the array codePhase[5] = {0}

    /* Find out the code phase for 5 satellites having maximum power */

17. Apply FFT to $x_{in}[n] \longrightarrow X[k]$, for n = 0 to N-1

18. **for sv**=01 to **sv**=32

    Initialize max = 0
    Initialize max_index = 0
    Multiply $X[k]$ and $C[\mathbf{sv}][k]$.
    $Y[k] = X[k] \cdot C[\mathbf{sv}][k]$, for k = 0 to N-1
    $R[n] = IFFT_k\{Y[k]\}$, for n = 0 to N-1, k = 0 to N-1
    for n=0 to **n** = N-1
        $R[\mathbf{n}] = R[\mathbf{n}] \times R^*[\mathbf{n}]$
        if R[**n**] > max:
            max = R[**n**]
            max_index = **n**
        end if
    end for

    /* Update max_power, visible_satellites_withMaxPower and codePhase arrays */

    for i = 0 to **i** = 4
        if max > max_power[**i**]
            for j = 4 to **j** = **i**-1
                max_power[**j**] = max_power[**j**-1]
                visible_satellites_withMaxPower[**j**] = visible_satellites_withMaxPower[**j**-1]
                codePhase[**j**] = codePhase[**j**-1]
                j = **j**-1
            end for
            max_power[**i**] = max
            visible_satellites_withMaxPower[**i**] = **sv**
            codePhase[**i**] = max_index
            break the loop
        end if
    end for

end for

/* Finding the Doppler shift for 5 satellites */

19. **for sv** = 0 to **sv** = 4

    Code phase $\hat{\tau}$ = codePhase[**sv**]
    Initialize max_of_max=0
    for $f_D = f_{min}$ to $f_D = f_{max}$ in f_{step} steps:
        Shift the signal $x[n]$ by f_D, for n = 0 to N-1

$$x_{sh}[n] = x_{in}[n + \hat{\tau}] \cdot e^{-j2\pi f_D n T_s} \tag{3}$$

        Initialize max_value = 0
        Initialize max_fd = 0
        Initialize z = 0
        for i = 0 to **i** = N-1
            z = z + x_{sh}[**i**] × c[visible_satellites_withMaxPower[**sv**]][**i**]
        end for
        z = Re(z)
        if (z > max_of_max)

max_of_max = z
max_fd = f_D
end if
end for
Doppler Frequency offset $f_{D_{sv}}$ = max_fd
end for