2025-9

Satellite Navigation Science and Technology for Africa

23 March - 9 April, 2009

Introduction to GPS Receiver Design Principals
(Part 6)

Ward Phillip. W.

Navward
U.S.A.

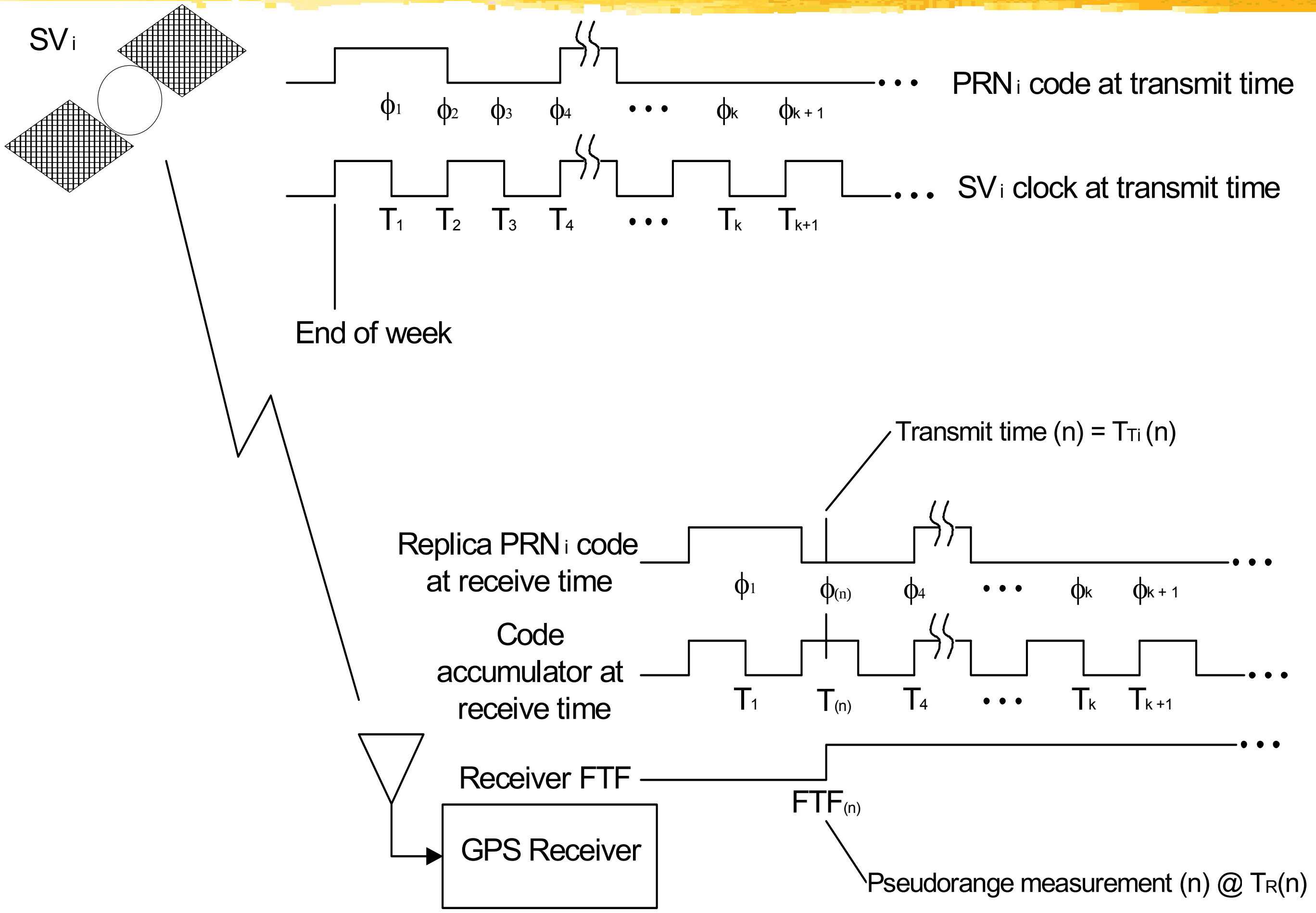
Session VI - Code
Extracting measurements from code and carrier loops

- Extracting measurements from code loop
  - Pseudorange definition
  - Satellite transmit time relationship to code phase
  - Pseudorange measurement
  - Relationship between PRN code generator and code accumulator
  - Measurement time skew
  - Code (time) accumulator
  - Maintaining the code accumulator
  - Obtaining a measurement from the code accumulator
  - Obtaining transmit time from C/A code
  - Synchronizing code accumulator to replica code generator
  - Adding a code setter to code generator
  - C/A-code setup
  - Obtaining transmit time from the C/A-code
  - GPS C/A-code timing relationships
  - GPS navigation message
  - Example of bit sync error in C/A-code

Pseudorange definition

- Pseudorange to SV_i where i is PRN number:
  - $PR_i(n)$ = c [$T_R(n)$ - $T_{Ti}(n)$](meters)
    where: c = GPS propagation constant
    = 2.99792458×10^8 (m/s)
  - $T_R(n)$ = receive time at epoch n of the GPS receiver's clock (seconds)
  - $T_{Ti}(n)$ = transmit time based on SV_i clock (seconds) of measurement at epoch n

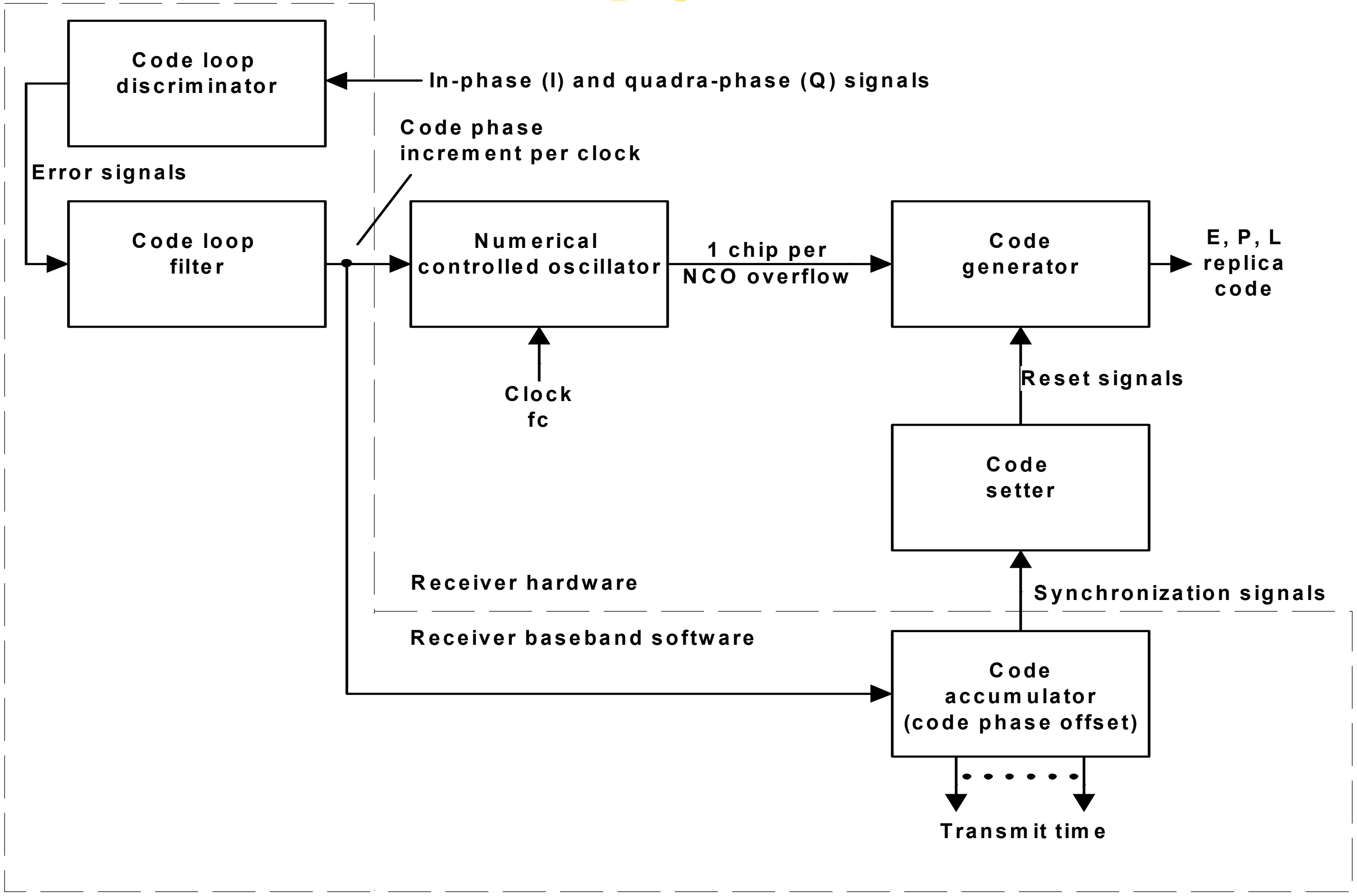
Satellite transmit time relationship to code phase



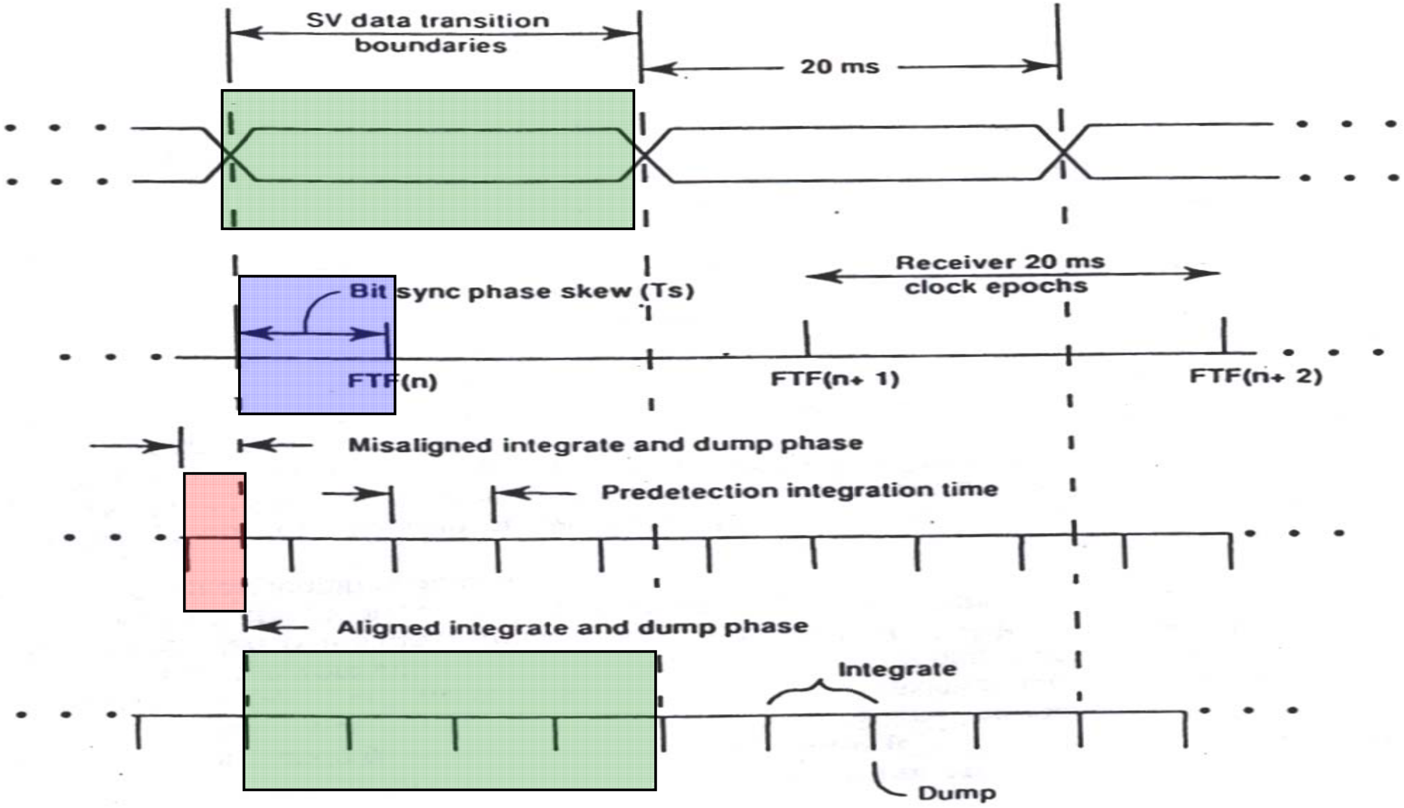
Pseudorange measurement

- Replica code state corresponds to receiver's best estimate of SV transmit time
  - Receiver knows replica code state because initial states are set during search process and it keeps track of code changes
  - Integer and fraction of chip replica code phase defined as code state
  - Receiver time keeper containing GPS time corresponding to replica code state defined as code accumulator

Relationship between PRN code generator and code accumulator



Measurement time skew



Code (time) accumulator

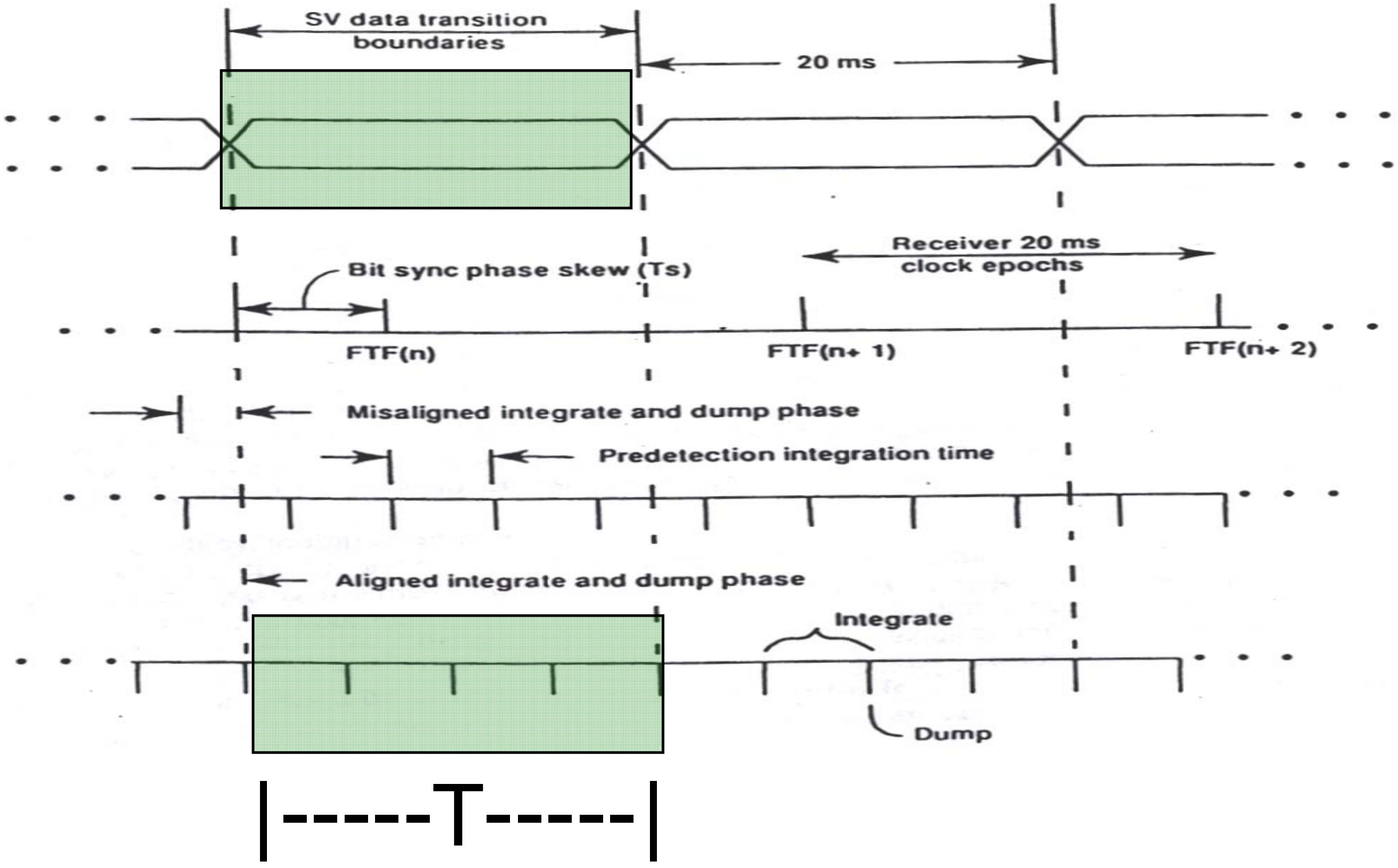
- Z-counter (19 bits)
  - Accumulates in GPS time increments of 1.5 s, then resets one count short of 1-week = 403,200. Max = 403,199
- X1-counter (24 bits)
  - Accumulates in GPS time increments of integer P chips, then resets one count short of 1.5 s = 15,345,000. Max = 15,344,999
- P-counter (same as NCO bits)
  - Accumulates in GPS time increments of fractions of one P-chip

Maintaining the code accumulator

- $P_{temp} = P + f_c \Delta\varphi_{CO} T$
- P = fractional part of P_{temp} (chips)
- $X_{temp} = (X1 + \text{whole part of } P_{temp}) / 15{,}345{,}000$
- $X1$ = remainder of X_{temp} (chips)
- Z = remainder of $[(Z + \text{whole part of } X_{temp}) / 403{,}200]$

  (1.5 seconds)
  - **where: P_{temp} = temporary P register**
  - **f_c = code NCO clock frequency (Hz)**
  - **$\Delta\varphi_{CO}$ = code NCO phase increment per clock cycle**
  - **= code NCO bias + loop filter Doppler correction, etc.**
  - **T = time between code NCO updates (seconds)**

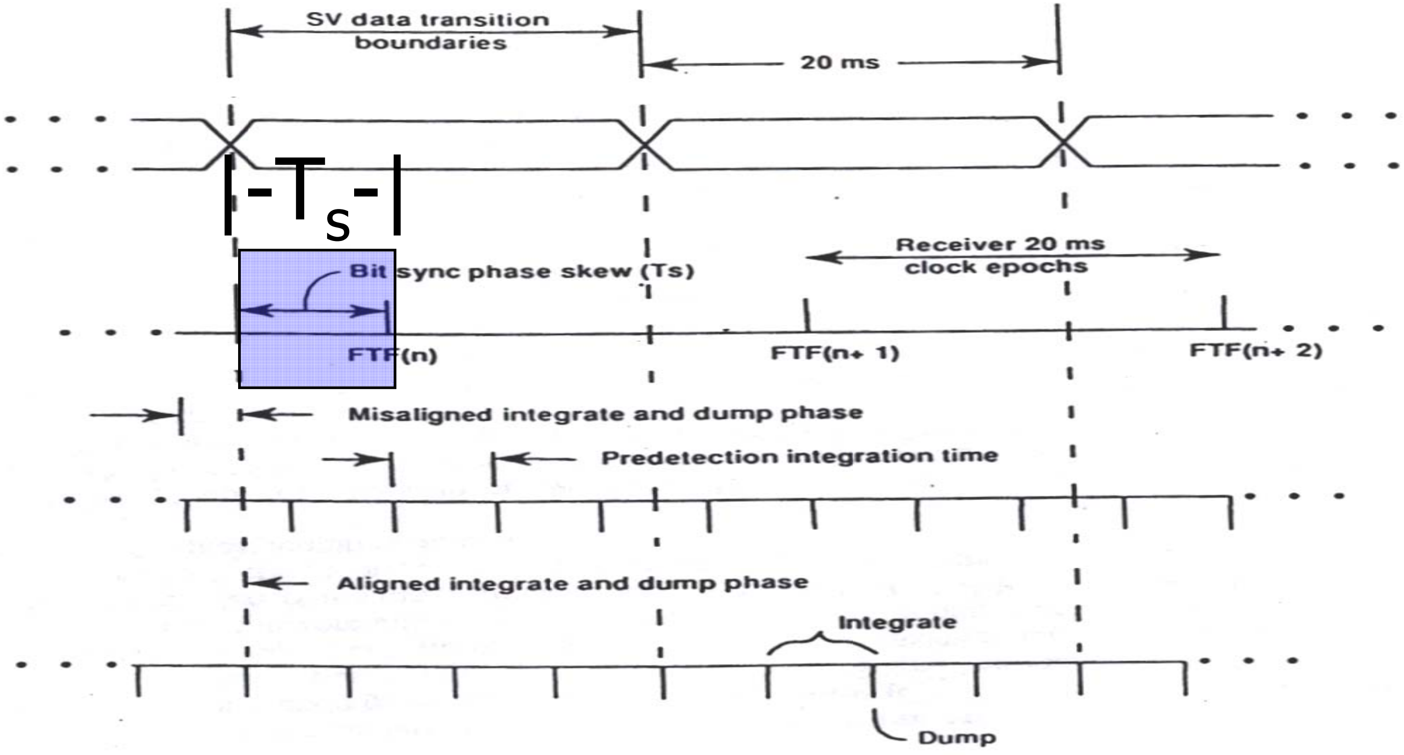
Maintaining the code accumulator with 20 ms PIT



Obtaining a measurement from the code accumulator

- P_{temp} = $P + f_c \Delta\varphi_{CO} T_s$
- P_i (n) = fractional part of P_{temp} (chips)
- X_{temp} = (X1 + whole part of P_{temp}) / 15,345,000
- $X1_i$ (n) = remainder of X_{temp} (chips)
- Z_i (n) = remainder of [(Z + whole part of X_{temp}) / 403,200] (1.5 seconds)
- T_{Ti} (n) = $[P_i (n) + X1_i (n)] / (10.23 \times 10^6) + Z_i (n) * 1.5$ (seconds)
- Note: code accumulator NOT changed

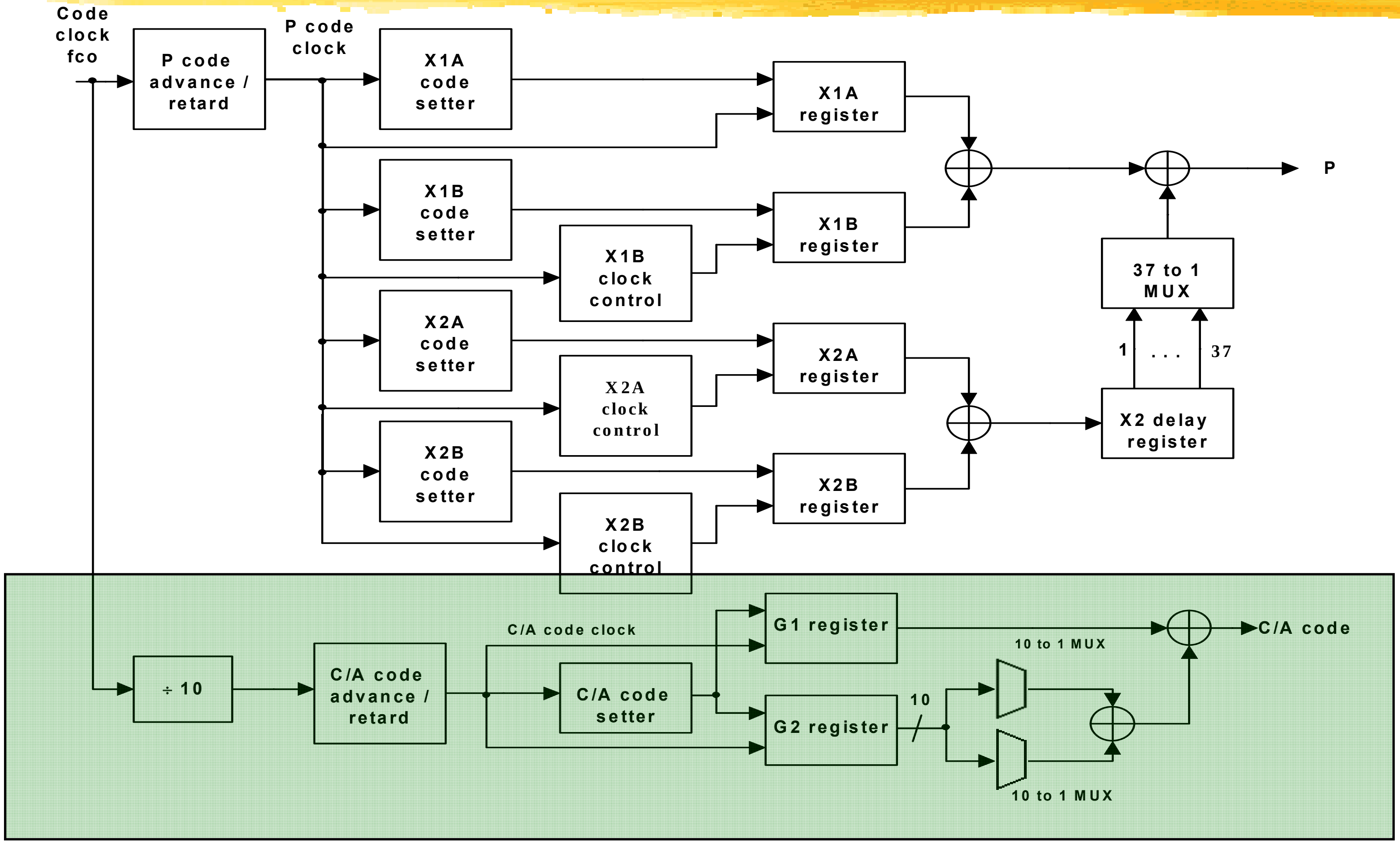
Obtaining a measurement from the code accumulator



Synchronizing code accumulator to replica code generator

- Most complicated part of process: Synchronizing code accumulator to C/A-code generator
  - Count sequences in code generator shift registers are pseudo random
  - Count sequences taking place in code accumulator are linear
- Reset timing events in PN shift registers are predictable

Adding a code setter to code generator



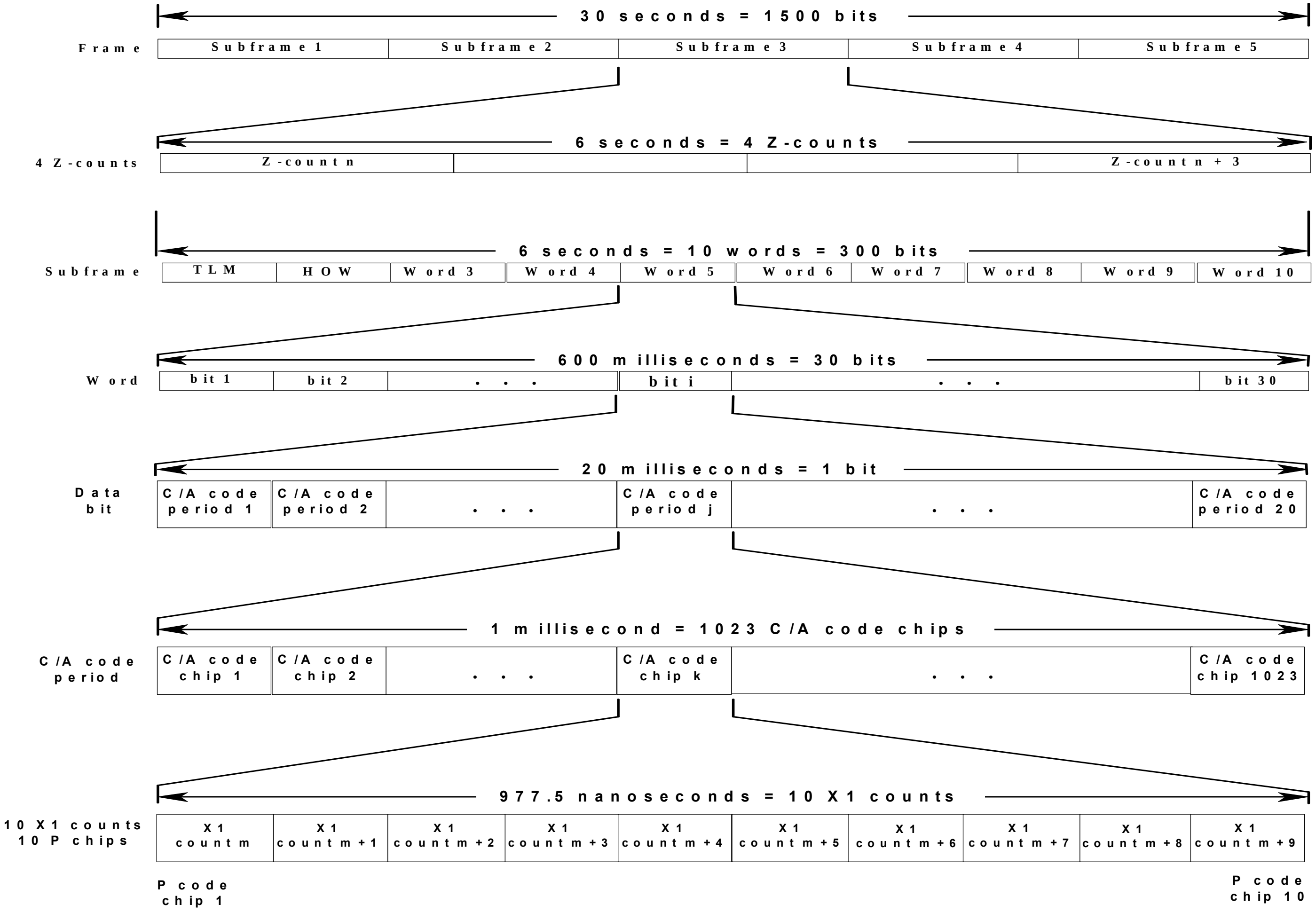
C/A-code setup

- Algorithm for code accumulator output to C/A-code setter:
  - G = remainder of [(whole part of {X1/10})/1023] where:
  - G = future scheduled C/A-code time value sent to the code setter
  - X1 = future scheduled GPS time of week in P chips ($0 \leq X1 \leq 15{,}344{,}999$)

Obtaining transmit time from the C/A-code

- C/A-code obtains transmit time from code accumulator in same manner as P(Y)-code
  - Difference is initialization of accumulator
  - Receiver reads Handover Word (HOW) after bit synchronization
  - Sets its code accumulator to correct Z-count at exactly right C/A-code epoch

GPS C/A-code timing relationships



30 seconds = 1500 bits

Frame: Subframe 1 | Subframe 2 | Subframe 3 | Subframe 4 | Subframe 5

6 seconds = 4 Z-counts

4 Z-counts: Z-count n | | | Z-count n + 3

6 seconds = 10 words = 300 bits

Subframe: TLM | HOW | Word 3 | Word 4 | Word 5 | Word 6 | Word 7 | Word 8 | Word 9 | Word 10

600 milliseconds = 30 bits

Word: bit 1 | bit 2 | . . . | bit i | . . . | bit 30

20 milliseconds = 1 bit

Data bit: C/A code period 1 | C/A code period 2 | . . . | C/A code period j | . . . | C/A code period 20

1 millisecond = 1023 C/A code chips

C/A code period: C/A code chip 1 | C/A code chip 2 | . . . | C/A code chip k | . . . | C/A code chip 1023

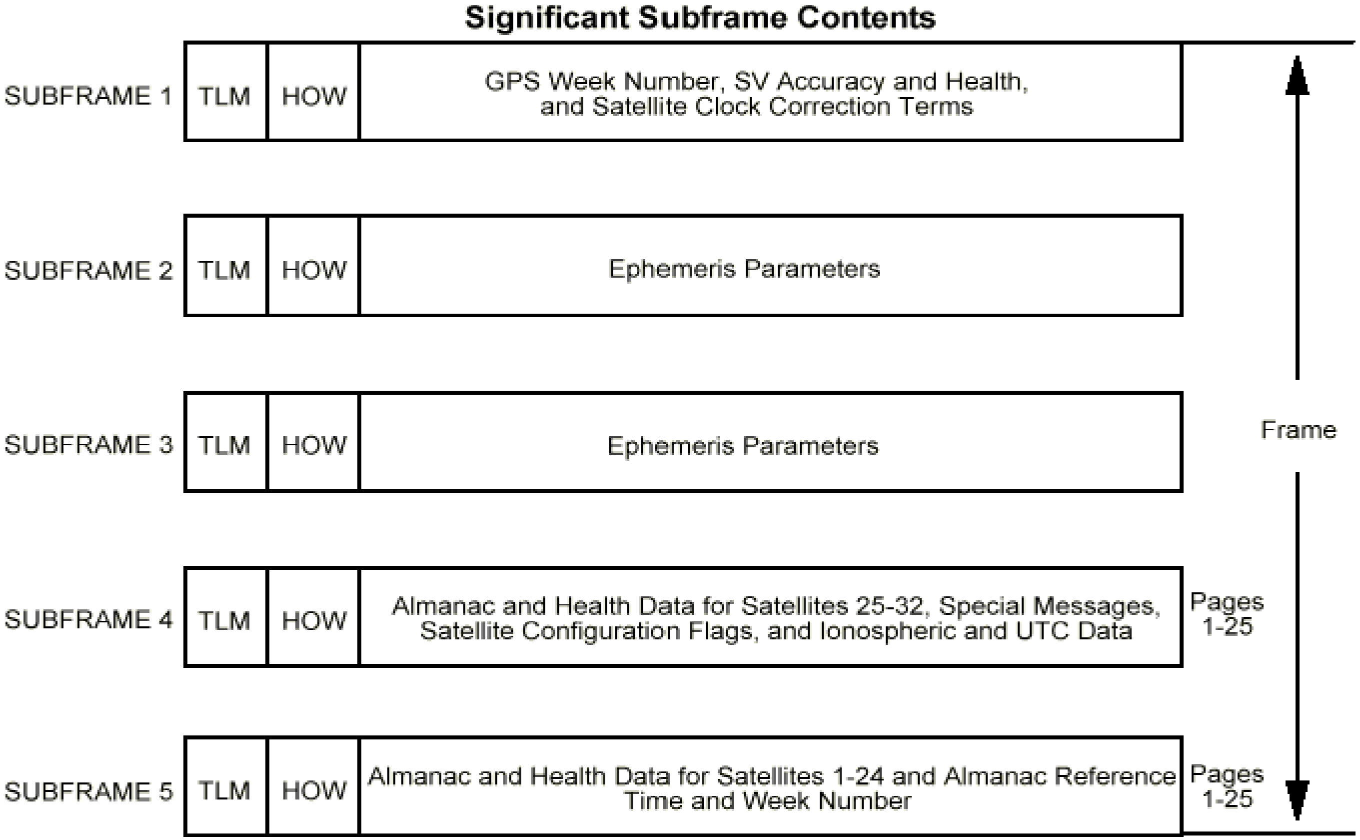
977.5 nanoseconds = 10 X1 counts

10 X1 counts
10 P chips: X1 count m | X1 count m+1 | X1 count m+2 | X1 count m+3 | X1 count m+4 | X1 count m+5 | X1 count m+6 | X1 count m+7 | X1 count m+8 | X1 count m+9

P code chip 1

P code chip 10

GPS navigation message format

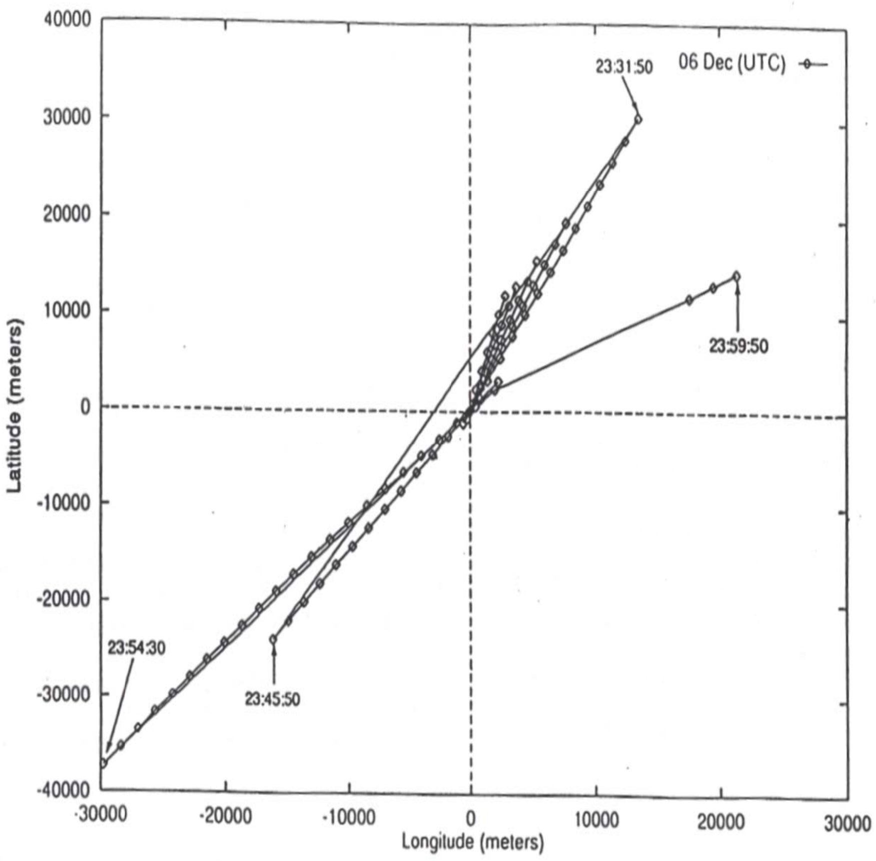


Significant Subframe Contents

SUBFRAME 1	TLM	HOW	GPS Week Number, SV Accuracy and Health, and Satellite Clock Correction Terms	
SUBFRAME 2	TLM	HOW	Ephemeris Parameters	
SUBFRAME 3	TLM	HOW	Ephemeris Parameters	
SUBFRAME 4	TLM	HOW	Almanac and Health Data for Satellites 25-32, Special Messages, Satellite Configuration Flags, and Ionospheric and UTC Data	Pages 1-25
SUBFRAME 5	TLM	HOW	Almanac and Health Data for Satellites 1-24 and Almanac Reference Time and Week Number	Pages 1-25

Each subframe is 300 bits (6 s @ 50 bps). Entire message repeats every 12.5 min (5 subframes × 300 bits/subframe × 25 pages = 37500 bits/message)

Example of bit sync error in C/A-code



Session VI - Carrier

Session VI - Carrier
Extracting measurements from code and carrier loops

- Extracting measurements from carrier loop
    - Maintaining the carrier accumulator
    - Obtaining a measurement from the carrier accumulator
    - Delta pseudorange measurement
    - Data demodulation
    - Bit error performance

Maintaining the carrier accumulator

- Carrier accumulator updated as follows:
  - $\Phi_{temp} = \Phi_{CA} + f_c \Delta\varphi_{CA} T$
  - Φ_{CA} = fractional part of Φ_{temp} (cycles)
  - $N_{CA} = N_{CA}$ + whole part of Φ_{temp} (cycles)
    where:
    - **Φ_{temp}= temporary Φ_{CA} register**
    - **f_C = carrier NCO clock frequency (Hz)**
    - **$\Delta\varphi_{CA}$ = carrier NCO carrier Doppler phase increment per clock epoch = carrier loop filter velocity correction + carrier loop velocity aiding (if any)**
    - **T = time between carrier NCO updates (seconds)**
    - **N_{CA} = integer number of carrier Doppler phase cycles since some arbitrary starting point**

Obtaining a measurement from the carrier accumulator

- The natural measurement obtained from the carrier accumulator associated with SVi is an ambiguous integer number of cycles, N_{CAi}, and an unambiguous fraction of a cycle (phase), Φ_{CAi}, defined at some epoch time
- Called an integrated carrier Doppler phase measurement
- To obtain integrated carrier Doppler phase measurement, N_{CAi} (n), Φ_{CAi} (n), for SV_i corresponding to carrier accumulator, propagate forward to nearest FTF(n) by skew time, T_s :
    - $\Phi_{temp} = \Phi_{CA} + f_c \Delta\varphi_{CA} T_s$
    - Φ_{CAi} (n) = fractional part of Φ_{temp} (cycles)
    - N_{CAi} (n) = N_{CA} + whole part of Φ_{temp} (cycles)

Delta pseudorange measurement

- Define delta pseudorange measurement, DPR_i (n), two integrated carrier Doppler measurements, ICD_i (n-K) and ICD_i (n), taken at FTF (n-K) and FTF (n)
  - FTF (n- K) occurs some integer number of FTFs earlier than FTF (n), ideally at PR measurements
  - ICD_i (n -K) = N_{CAi} (n-K), Φ_{CAi} (n-K)
  - ICD_i (n) = N_{CAi} (n), Φ_{CAi} (n)
  - DPR_i (n) = [ICD (n) – ICD (n-K)]λ (meters)
  - Λ = 0.1903 m/cycle at L1 (and 0.2442 m/cycle at L2)

Data demodulation

Data bits are estimated from 20 ms correlator outputs as:

$$\hat{b}_k = \mathrm{sign}(I_{Pk})$$

This is ordinary BPSK data demodulation. The well-known expression for BPSK bit error rate is:

$$P_b = \frac{1}{2}\mathrm{erfc}\left(\frac{E_b}{N_0}\right)$$

The ratio of bit energy E_b to noise power density N_0 may be related to P/N_0 using:

$$E_b = P/\underset{\uparrow}{R_b}$$

data rate (50 Hz for GPS)

Bit error performance