GLAST Large Area Telescope

Instrument Flight Software Development Team

Functional Demonstration
February 18, 2005

Stanford Linear Accelerator Center

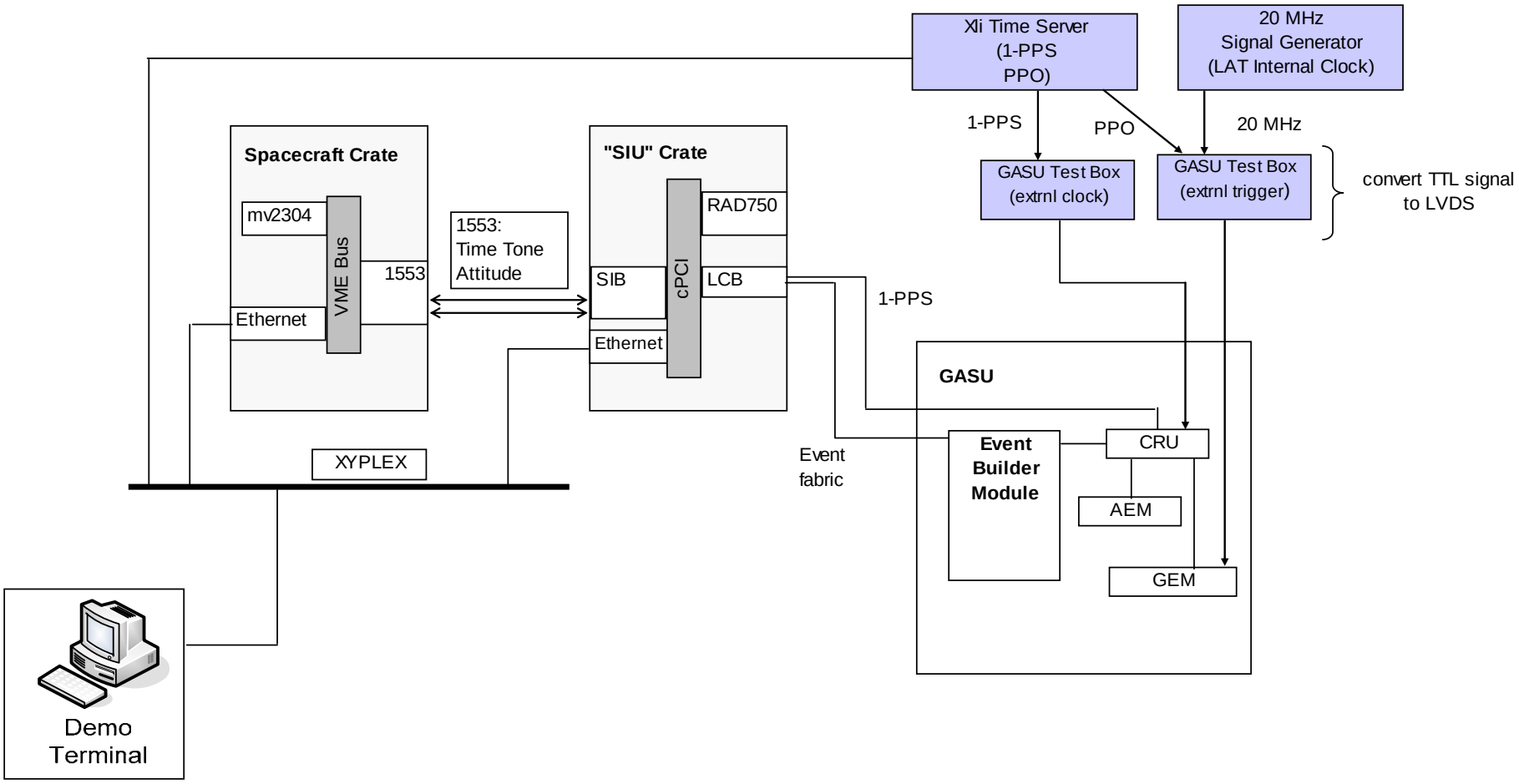
Demonstration Overview: Time Hack Processing

- **Today's demo covers Time Processing:**
  - **The demo uses all of the real time signals to construct time tables:**
    - **Spacecraft 1-PPS signal (from an Xli time server, but properly routed via GASU)**
    - **Spacecraft time tone message over 1553**
    - **"GEM" 20 MHZ clock signal (produced by a signal generator, routed via GASU)**
  - **Values from the time table are used by the Attitude Processing software, also demonstrated today**
  - **Addresses the following requirements:**
    - **5.3.4.1 GPS Time Hack from SC**
    - **5.3.4.2 GPS Time Hack Integrity Check**
    - **5.3.4.4 LAT Clock Correlation**
    - **5.3.4.5 Time Stamp Accuracy**

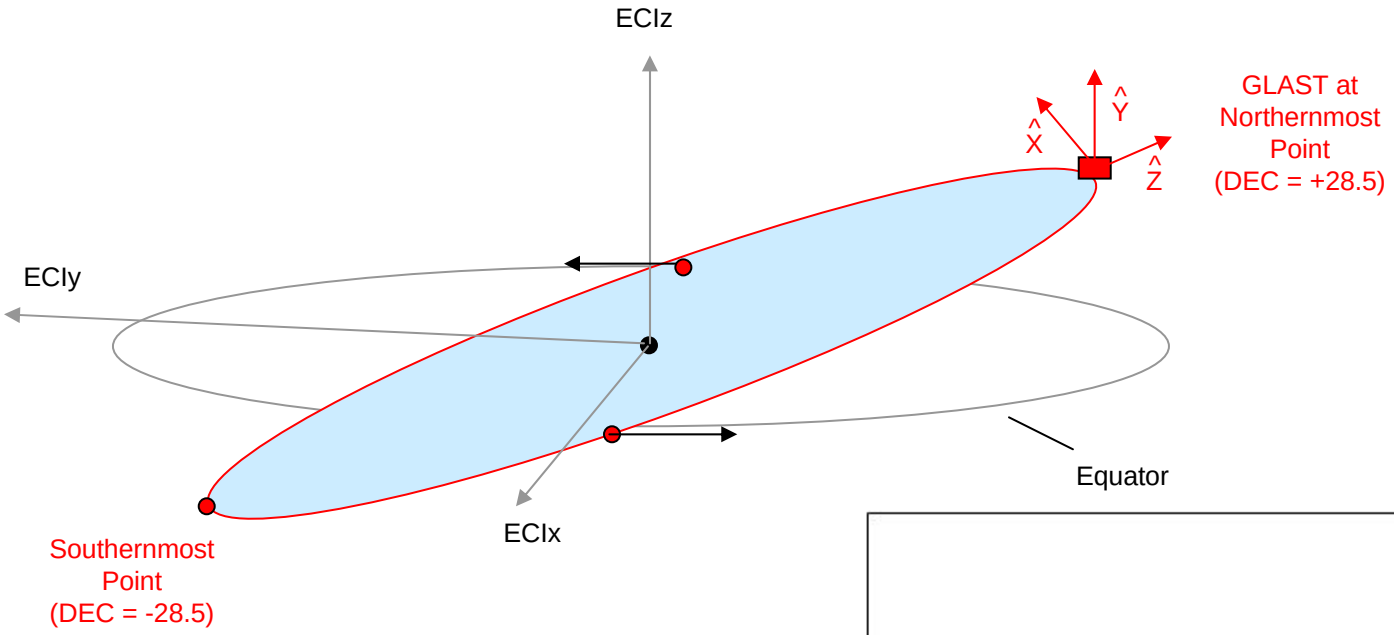
Demo Overview: Attitude Processing

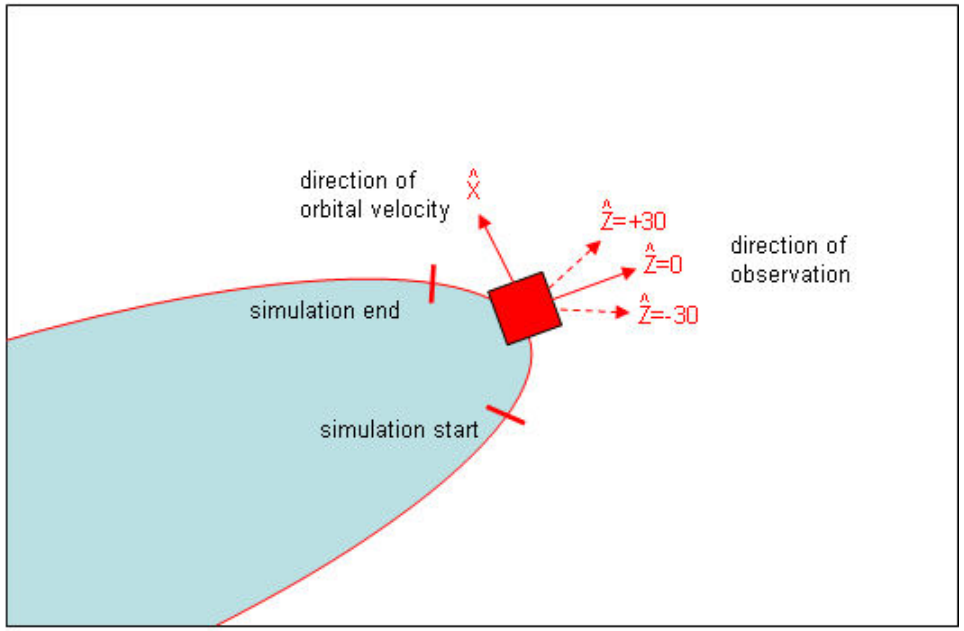
- **And Attitude Processing**
  - **Using sets of attitude simulation data as source data, demo shows that the attitude processing software:**
    - **Receives the attitude records from the Spacecraft**
    - **Given a timestamp (read from the Time Processing software) and an attitude vector in LAT coordinates, converts the vector into J2000 coordinates**
  - **Addresses the following requirements:**
    - **5.3.11.3.3 Process Attitude Data**
    - **5.4.2.1 LAT Coordinate System**
    - **5.4.2.2 Observatory Coordinates**
    - **5.4.2.3 Celestial Coordinate System**

Context for the Demonstration



Visualizing the Vectors Used in Attitude Processing





Outputs of the Attitude Demo

- **As the demo proceeds, and the Spacecraft moves towards the northernmost point in the simulation, a series of records like the two shown below are generated**
  - **Time records and transformed attitude vectors (Cartesian and RA/DEC) are shown**
  - **Vector dot and cross products are shown to confirm proper behavior of the algorithms**

att_demo - Notepad

File Edit Format View Help

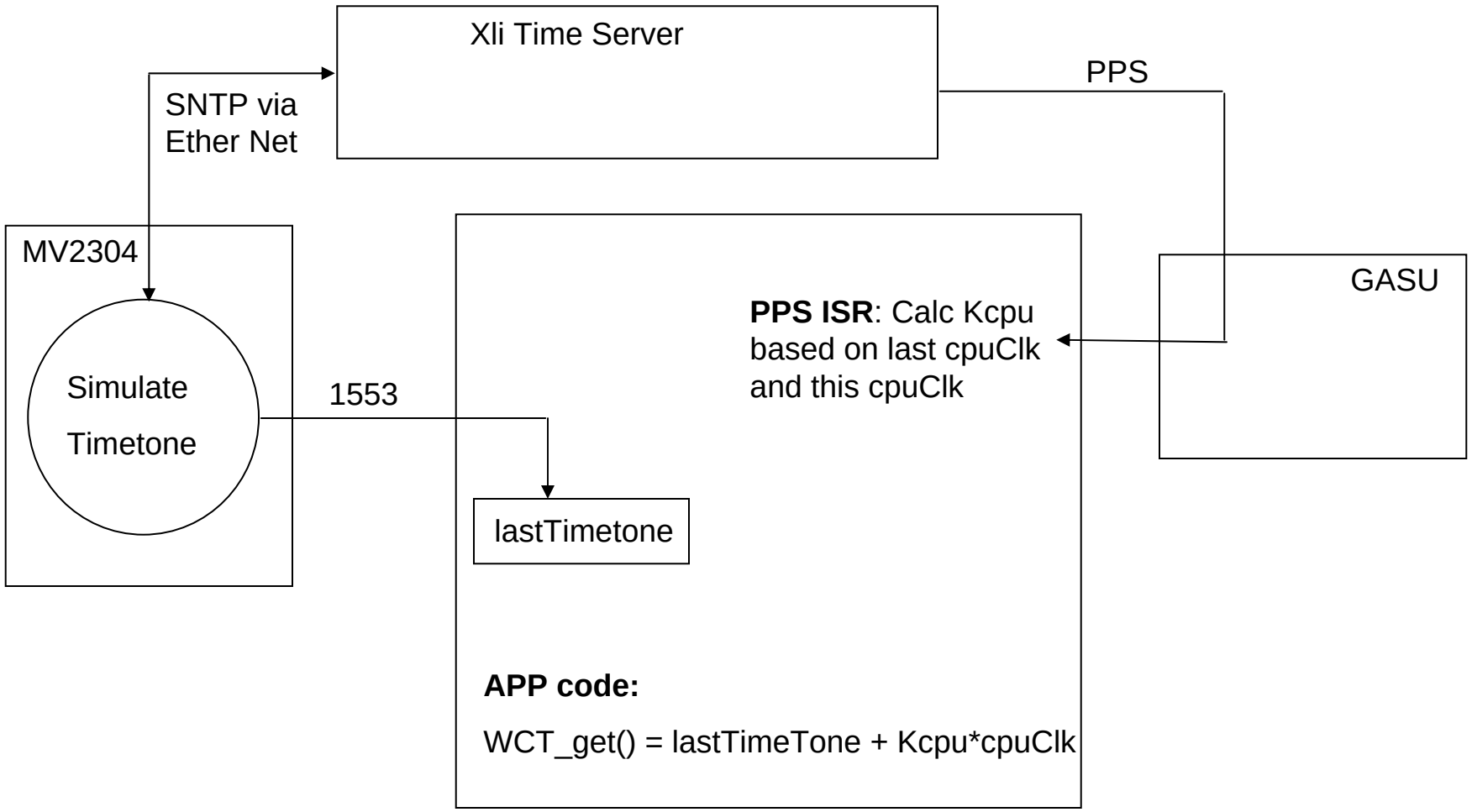
```
------------------------------------
Time: 234014410000000000
Att bottom: 234014401600000000
Att top:    234014409400000000
Z axis (rect): x=0.910067 y=0.020292 z=0.413965
Z axis (sphr): RA=1.277319 DEC=24.454160
X axis (rect): x=-0.129411 y=0.962777 z=0.237305
X axis (sphr): RA=97.655474 DEC=13.727535
Y axis (rect): x=-0.393741 y=-0.269535 z=0.878818
Y axis (sphr): RA=214.393566 DEC=61.500065
Pos X/Y (rect): x=-0.369924 y=0.490196 z=0.789218
Pos X/Y (sphr): RA=127.039819 DEC=52.112484
Neg X/Y (rect): x=0.186909 y=0.871376 z=-0.453618
Neg X/Y (sphr): RA=77.893557 DEC=-26.976045
Pos X/Z (rect): x=0.552007 y=0.695135 z=0.460518
Pos X/Z (sphr): RA=51.546872 DEC=27.420506
Neg X/Z (rect): x=0.735022 y=-0.666438 z=0.124918
Neg X/Z (sphr): RA=317.801681 DEC=7.175994
X * Z = -0.000000
Y * Z = 0.000000
X * Y = 0.000000
(Pos X/Y) * (Neg X/Y) = 0.000000
(Pos X/Z) * (Neg X/Z) = 0.000000
X x Y: x=0.910068 y=0.020292 z=0.413965
```

att_demo - Notepad

File Edit Format View Help

```
------------------------------------
Time: 234014410000000000
Att bottom: 234014401600000000
Att top:    234014409400000000
Z axis (rect): x=0.910067 y=0.020292 z=0.413965
Z axis (sphr): RA=1.277319 DEC=24.454160
X axis (rect): x=-0.129411 y=0.962777 z=0.237305
X axis (sphr): RA=97.655474 DEC=13.727535
Y axis (rect): x=-0.393741 y=-0.269535 z=0.878818
Y axis (sphr): RA=214.393566 DEC=61.500065
Pos X/Y (rect): x=-0.369924 y=0.490196 z=0.789218
Pos X/Y (sphr): RA=127.039819 DEC=52.112484
Neg X/Y (rect): x=0.186909 y=0.871376 z=-0.453618
Neg X/Y (sphr): RA=77.893557 DEC=-26.976045
Pos X/Z (rect): x=0.552007 y=0.695135 z=0.460518
Pos X/Z (sphr): RA=51.546872 DEC=27.420506
Neg X/Z (rect): x=0.735022 y=-0.666438 z=0.124918
Neg X/Z (sphr): RA=317.801681 DEC=7.175994
X * Z = -0.000000
Y * Z = 0.000000
X * Y = 0.000000
(Pos X/Y) * (Neg X/Y) = 0.000000
(Pos X/Z) * (Neg X/Z) = 0.000000
X x Y: x=0.910068 y=0.020292 z=0.413965
```

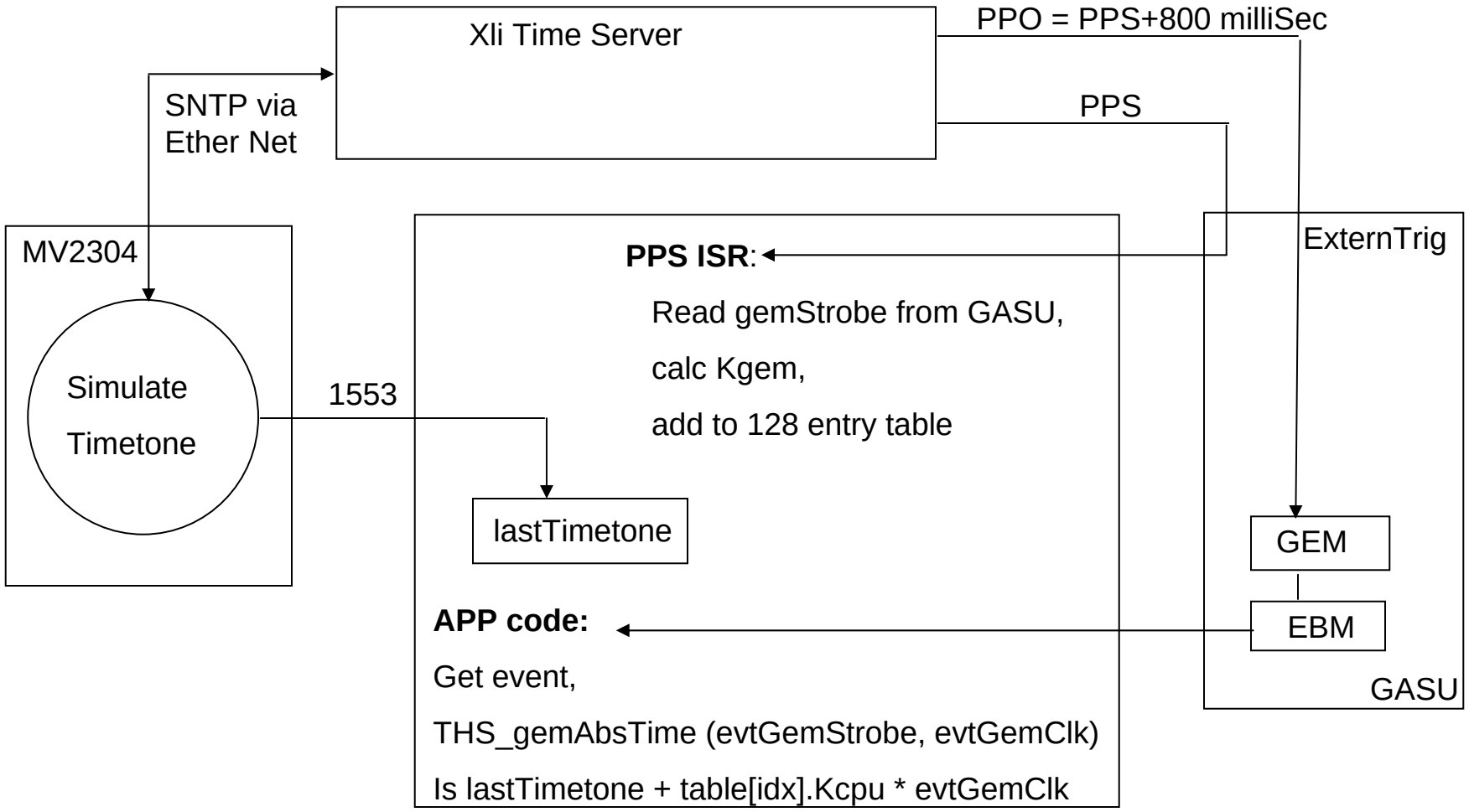
CPU Time Hack (CPU PCI clock 8.3/2.08 MHz)



CPU Time Hack Test

- Setup wakeup timer on one second interval (of corrected time)
- On every wakeup
  - Read cpuClk and calculate delta cpuCLk
  - Determine how close to expected value
  - In this case, a short test, we don't expect any clock drift so the delta cpuClk should be close to the same every second

GEM Time Hack (20 MHz LatP Clock)



GEM Time Hack Test

- **Setup GASU to trigger event at PPS+800 milliSec**
- **On each Event**
  - **Calculate absTime of event from evtGemStrobe and eventGemClk in the event packet**
  - **Uses table of last 128 sec to get Kcpu for time calculation**
  - **Expect event time to be xxxx.800000**