GPS IF Core Specification

Rev v0.85

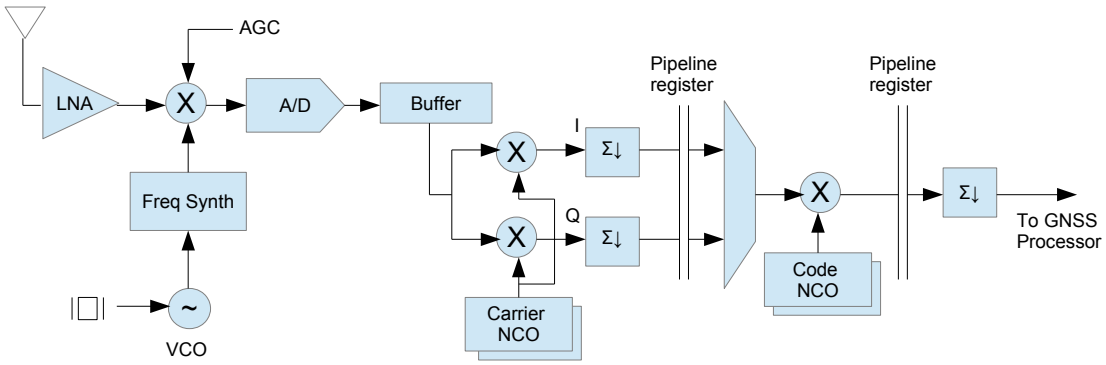
Abstract

CoreSemi's GPS IF Core is a component of the larger technology platform, designed to enable accurate time tracking from satellite signals, in which an antenna is connected to a digital GPS/GNSS receiver via a synchronous communication channel.

The antenna converts the received satellite signal to digital IF. Digital samples are captured into a local clock domain. The digital GPS/GNSS receiver includes a master clock driven by a temperature corrected oscillator and corrected to the satellite time. Signal processing obtains the satellite time data and the master clock is corrected based upon the satellite time data and the time stamp.

The Open Hardware release of the GNSS receiver is a patent free subset of the CoreSemi GNSS Time receiver, which also contains a digital antenna interface design.

Figure 1, below, depicts a single channel of the GPS receiver. It should be noted that the receiver consists of up to 7 such channels, time sliced.



Interface List and Description

GPS IF has external interface groups and a small set of global signals shared with the CPU, and other assets in the technology platform. These interfaces are listed below for reference.

Table 1. GPS IF Interfaces

Name	Type	Description
GPS IF TOP		
clk	in; std_logic	clock for internal operations
rst	in; std_logic	HW reset signal
gps_clk	in; std_logic	gps clock, input from IO pin
gps_d	in; std_logic vector	(2 bits), gps data, input from IO pin
ppsdds	in; std_logic	parts per second, for GPS IF time module
blgps	in, blgps_t	control signals for GPS over bitlink of GPS IF time module
dma	out; dma_reg_t	Direct report signal to DMAC (bit 6-0)
intrpt	out; std_logic	interrupt signal generated by GPS IF time module
bi	in; bist_scan_t	built-in self test scan input
bo	out; bist_scan_o	built-in self test scan output
ring_i	in; rbus_9b	ring bus input (word, stall)
ring_o	out; rbus_9b	ring bus output (word, stall)
bus_clk	in; std logic	clock for bus interface (N x slower than clock, N = natural number)
db_i	in; cpu_data_o_t	cpu bus input (address, data, enable, read, write)
db_o	out; cpu_data_i_t	cpu bus output (data, acknowledge)

Table 2. Address Map

32 bit Address	Description
abcc0000	Base address on CPU bus
abcc0000 + ch	C/A code shift, PNCO inc, DMA channel
abcc0020 + ch	G2 shift & Acquisition mode, Channel activation
abcc0040 + ch	PNCO
abcc0060	SW Reset
abcc + 8*ch + N	Output Buffer (N: 0-5 for EI,EQ,PI,PQ,LI,LQ)

32 bit Address	Description
abcc0200	GPSIF status
abcc0204	Input buffer from CPU bus

Bit Assignments

1. S/W RESET

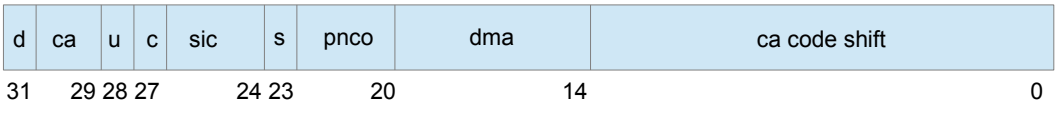
bit 31		0	
bit 8	“s” sample bit mode	0	2 bit sample
		1	1 bit sample
bit 1-0	“mn” input mode	00	I/O port
		01	CPU bus
		1x	Reserved

Notes:
- Currently, other bits are used to dump signals for debug on FPGA
- ’s’ represents sample bit mode of GPS over bitlink

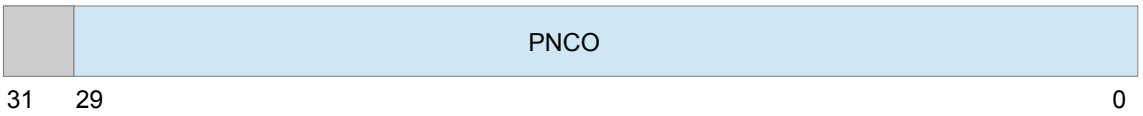
2. CHANNEL INITIALISATION

2.1. C/A code shift

(direct, increment/plus/minus), PNCO Increment, DMA channel



bit 31	“d” direct write of C/A code shift & DMA channel	0	disable
		1	enable
bit 30-29	“ca” C/A code shift update type: NOP, INC, PLS, MNS		
bit 28	“u” PNCO update	0	disable
		1	enable
bit 27	“c” shift increment code set	0	disable
		1	enable
bit 26-24	“sic” shift increment code	000	+ 1
		001	+ 8
		010	+ 16
		011	+ 24
		100	+ 4
		101	- 24
		110	- 16
		111	- 8
bit 23	“s” PNCO increment code set	0	disable
		1	enable
bit 22-20	PNCO increment code	000	1
		001	2
		010	4
		011	8
		100	16
		101	32
		110	64
		111	128
bit 19-14	DMA channel		DMA channel
bit 13-0	C/A code shift		direct write



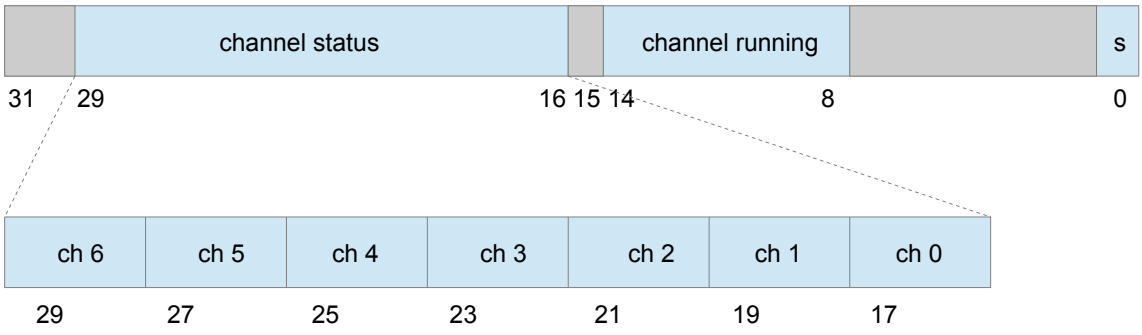
2.2 PNCO Direct write

bit 31-30	don't care		ignored by HW
bit 29-0	PNCO direct write		30 bit signed

2.3. G2 Shift & Acquisition mode

bit 31	"a" acquisition mode	0	disable
		1	enable
bit 30-10	don't care		ignored by HW
bit 9-0	G2 shift		10 bits

2.4 GPS IF status register



bit 31-30	don’t care		ignored by HW
bit 29-28	ch 6 status	00	IDL
		01	ACQ
		10	RDY
		11	VLD
bit 27-26	ch 5 status	nn	as above
bit 25-24	ch 4 status	nn	as above
bit 23-22	ch 3 status	nn	as above
bit 21-20	ch 2 status	nn	as above
bit 19-18	ch 1 status	nn	as above
bit 17-16	ch 0 status	nn	as above
bit 14	ch 6 running	0	ready
		1	run
bit 13	ch 5 running	0/1	as above
bit 12	ch 4 running	0/1	as above
bit 11	ch 3 running	0/1	as above
bit 10	ch 2 running	0/1	as above
bit 9	ch 1 running	0/1	as above
bit 8	ch 0 running	0/1	as above
bit 7-1	don't care		ignored by HW
bit 0	“s” input buffer status	0	busy
		1	fill

Software Control

1. S/W reset

S/W reset initialises GPS IF:

- Input mode is specified by bit 31, and 0 & 1 correspond to from I/O port & CPU bus.

- All channels are set to IDL.
- Input buffer is cleared by initialising read scope & write pointers.
- Reset 1-ms counter that counts number of data of 1023*16 for 1 ms; the counter consists of common higher 11-bit one that skips 1023*2 and 1023*2+1, and lower 3-bit one just to count 8 for each 8 data processing.)

2. Channel initialisation and updates

For Tracking:

1. Direct write of C/A code shift of acquired satellite, the value is the 1-ms counter value corresponding to the last data of the 1-ms period of the acquired satellite.
2. Direct write of PNCO of the acquired satellite
3. Write G2 shift of the acquired satellite w/ disabling acquisition; write makes the channel RDY; the initialised channel starts running after the higher 11-bits of the C/A code shift and the 1-ms counter match.
4. Write PLS/MNS mode if C/A code shift is to be adjusted with +/-1; H/W adjust the shift as early as possible w/ keeping carrier NCO phase.
5. Write new PNCO if it is to be adjusted; H/W changes the PNCO as early as possible. The carrier NCO phase is naturally kept.

For Serial Acquisition (Not use acquisition mode):

1. Direct write of initial C/A code shift of each channel
2. Direct write of initial PNCO
3. Write of G2 shift of a satellite w/ disabling acquisition. The write makes the channel RDY.
4. Increment PNCO to scan Doppler shift range of +-10 kHz; H/W initialises carrier NCO phase to 0 when it increment the PNCO. Each evaluation then starts w/ the same carrier NCO phase.
5. Increment C/A code shift to scan all the shift range; H/W increment the PNCO and C/A code shift at the 1-ms boundary of the channel. So, the control must be done during the last 1-ms evaluation of the channel.)
6. Write of G2 shift of another satellite w/ disabling acquisition.

For FFT Acquisition (Use acquisition mode):

Note: GPS IF Core outputs I & Q results of Carrier NCO Multiply & Accumulate of every eight data for FFT Acquisition to ring bus; DSP required for remaining processing.

1. Direct write of initial PNCO
2. Set acquisition mode enable. The write makes the channel RDY. (G2 shift is not used.)
3. The initialised channel starts running when the 1-ms counter becomes 0.
4. Increment PNCO to scan Doppler shift range of +-10 kHz

3. Data input

From I/O port

1. Write data w/ setting bit 31 to be 0 to S/W reset address; default input mode is to use I/O port. Nothing is to be done after H/W reset.

From CPU bus

1. Write data w/ setting bit 31 to be 1 to S/W reset address.
2. CPU bus master reads GPSIF status register to check input buffer status BUSY or not, and wait until the status becomes FILL.
3. After it becomes FILL, the CPU bus master writes 16*32 bits of data.
4. Input cycles must be more than 56 (7ch*8cycles) per 16 CPU bus operations for 7 channel case; It is safe to write data every 4 cycles or less. (4*16 = 64 > 56)

4. Read evaluation result

By CPU bus master:

1. CPU bus master checks each channel status by reading GPSIF status register.
2. If the status is VLD, it reads valid results from output buffers.
   - The results of each channel are six (6) data: EI,EQ,PI,PQ,LI, and LQ.
   - H/W changes the status to RDY after reading the LQ.
   - LQ should therefore be read last.
   - GPS IF has output buffers that is separated from accumulating buffers.
   - Then, the results in the output buffers are valid for 1 ms.
   - H/W set the fail flag of each channel when H/W observes a read access while the status is RDY (read-before-write), or H/W writes a new result while the status is VLD (write-before-read). Currently, the fail bits are not assigned to GPSIF status register.

By DMAC:

1. When DMAC receives the DMA channel number from GPSIF, it reads the results.
2. GPSIF outputs the DMA channel assigned to each channel to DMAC when the results of the channel become valid; DMAC therefore does not have to check the output validity.

Hardware Behaviour

1. H/W reset

H/W reset initialises GPSIF. Default input mode is from I/O port. The others are the same as the SW reset.

2. Input buffer write

From I/O port, GPSIF has a buffer consisting of 4 sets of 8 data. The write uses a special clock synchronised to the I/O port. So, the write pointer and write enable signal are clocked by the special clock. The GPSIF processing is faster than the input write. So, the write is always possible. The higher 2 bits of the write pointer is passed to main body of the GPSIF. For an asynchronous pass, the 2 bits changes with a sequence of 00 => 01 => 11 => 10 => 00. Then, the 2-bit code is always correct even when the signal change timing is slightly different. (Avoiding 01 => 10 or 10 => 01 is important.)

From CPU bus, the GPSIF has a buffer consisting of 32 sets of 8 data assuming a 512 bit packet for the data. GPSIF clears the input buffer status bit of the GPSIF status register to show it is BUSY when the write pointer returns to 0, until read scope pointer points the last scope. CPU bus master must wait until the bit is set for writing new input data. Since the last scope consists of the last and

first 8 data and the processing of the scope requires 7*8 = 56 cycles, and 512 bit buffer filling takes 512/32 = 16 CPU bus operations, the bus master must not issue the last input data write within the 56 cycles. Is is safe to write data every 4 cycles or less. Assuming future data input from ring bus, it takes 512/8 = 64 ring bus cycles or more, and no restrictions are necessary to keep the input data.

3. Input buffer read by GPSIF H/W

Input buffer is divided into multiple sets of 8 data, and two consecutive sets are specified by the read scope pointer as a current input data scope. Then, a 4-bit read pointer that is lower 4 bits of C/A code shift of each channel can specify any 8 data in the scope. So, all the channels can share the input buffer with different C/A code shift. However, this is not true when the C/A code shift changes. The direct write is for initialisation, and it must be done while the channel is IDL. The change by INC/PLS mode requires a scope skip to ignore data between the last and new 1 ms. For negative INC, HW skips data of 1022*16 + (negative)INC. If the new shift can use the same scope, HW does not skip the scope. For MNS mode, HW uses the same scope twice if the scope is changed. If the read scope reaches to the writing entry of the input buffer, GPSIF stall processing of the data.