Features

- 12 channels (or user defined)
- 6 correlator arms per channel
- Compatible with Zarlink GP2015 RF front end
- Verilog or VHDL versions

Overview

The Namuru tracking module provides tracking and measurement services to navigation firmware running on a micro-processor. The Namuru tracking module is well-suited for integrating location functionality into a device with an FPGA chip. The inherent flexibility of the FPGA solution provides advantages in the changing landscape of GNSS, and facilitating easy application specific modifications.

Functional Description

The design can be partitioned into three main blocks as seen in Fig. 1. There are a number (nominally 12) of identical tracking channels, a time-base block and a register and address decoder block for interfacing with a micro-processor.

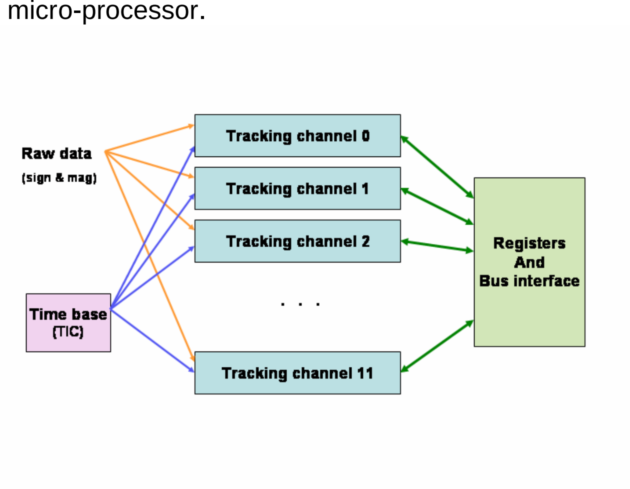

Figure 1. Tracking module block diagram

Time-base

The time-base provides two primary signals; the TIC, and the accumulator interrupt. The TIC signal synchronizes measurements across all channels. The accumulator interrupt provides an interrupt to the micro-processor so that accumulation and measurement data can be collected and processed. The time-base also provides a 40/7 MHz sample clock to the RF front end.

Register and Bus Interface Block

The tracking module attaches to a processor as a memory-mapped peripheral. In the test-bench system this is a NiosII soft core processor and the attaching process is handled by the SOPC builder in the 'Quartus' software from Altera.

The tracking module occupies a 256 byte address space, however many spare slots exist. The address map is provided in tables 1 to 4. Explanations of the register functions can be found in the following sections.

Function	address offset (hex)	block
Channel 0 block	00 - 0F	CH0
Channel 1 block	10 – 1F	CH1
Channel 2 block	20 – 2F	CH2
Channel 3 block	30 – 3F	CH3
Channel 4 block	40 – 4F	CH4
Channel 5 block	50 – 5F	CH5
Channel 6 block	60 – 6F	CH6
Channel 7 block	70 – 7F	CH7
Channel 8 block	80 – 8F	CH8
Channel 9 block	90 – 9F	CH9
Channel 10 block	A0 – AF	CH10
Channel 11 block	B0 - BF	CH11
	C0 - CF	spare
	D0 - DF	spare
Status block	E0 - EF	status
Control block	F0 - FF	control

Table 1. Address Map

Channel block	address offset (hex)	register function
write	0	PRN_KEY
write	1	CARRIER_NCO
write	2	CODE_NCO
write	3	CODE_SLEW
read	4	I_EARLY
read	5	Q_EARLY
read	6	I_PROMPT
read	7	Q_PROMPT
read	8	I_LATE
read	9	Q_LATE
read	A	CARRIER_MEASUREMENT
read	B	CODE_MEASUREMENT
read	C	EPOCH
read	D	EPOCH_CHECK
write	E	EPOCH_LOAD
	F	spare

Table 2. Tracking channel offsets

Status block	address offset (hex)	register function
read (cleared by read)	0	STATUS
read (cleared by read)	1	NEW_DATA
read	2	TIC_COUNT
read	3	ACCUM_COUNT
	4 - F	spare

Table 3. Status block offsets

Control block	address offset (hex)	register function
write	0	RESET
write	1	PROG_TIC
write	2	PROG_ACCUM_INT
	3 - F	spare

Table 4. Control block offsets

Tracking Channel

A block diagram of the tracking channel is given in figure 2. The primary blocks are the carrier_mixers, code_mixers, accumulators, carrier_nco, code_nco, code_generator and the epoch counter.

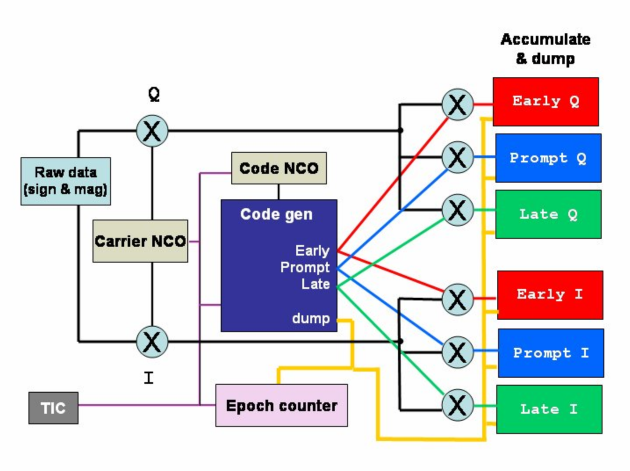

Figure 2. Tracking channel block diagram

There are quadrature and in-phase arms for early, prompt and late accumulators. The early – late separation is one chip.

Tracking Channel Registers

The **PRN_KEY** register is used to set the gold code sequence for that channel. The key is a 10 bit initial value that is loaded into the G2 linear feedback shift register in the code generator. PRN key writes have immediate effect. The PRN keys are shown in table 3, in hexadecimal.

1	3EC	11	2E8	21	260	31	158
2	3D8	12	3A0	22	0C0	32	2B0
3	3B0	13	340	23	0CE	33	160
4	360	14	280	24	270	34	0B0
5	096	15	100	25	0E0	35	316
6	12C	16	200	26	1C0	36	22C
7	196	17	226	27	380	37	0B0
8	32C	18	04C	28	300		
9	258	19	098	29	056		
10	374	20	130	30	0AC		

Table 5. PRN keys

The **CARRIER_NCO** register is used to control the frequency of the local carrier oscillator. It has a 30 bit wide accumulator. Writes to the carrier NCO register have an immediate effect on the frequency. The nominal center frequency is 1.405396825MHz.

f = p X clk / 2^30, where:

- f is the NCO frequency
- p is the register value
- clk is the clock frequency

For a 40MHz clock, p = 0x23FA689 for the nominal center frequency.

The resolution (increment value) is 0.037252902Hz

The **CODE_NCO** register is used to control the frequency of the locally generated C/A code sequence. It has a 29 bit wide accumulator. Writes to the code NCO register have an immediate effect on the frequency. The code NCO operates at twice the C/A code chipping rate of 1.023MHz. This is to enable half-chip spacing between the early-prompt and late code sequences.

Code f = 0.5 X p X clk / 2^29, where:

- Code f is the code frequency
- p is the register value
- clk is the clock frequency

For a 40MHz clock, p = 0x1A30552 for the center frequency of 1.023MHz.

The resolution (increment value) is 0.074505805Hz.

The **CODE_SLEW** register controls the delay of the C/A code sequence at the start of the sequence. A write to this register will delay the start of the C/A sequence at the next dump. The range is 1 to 2045 half chips.

I_EARLY is the accumulation value for the in-phase, early correlator arm.

Q_EARLY is the accumulation value for the quadrature, early correlator arm

I_PROMPT is the accumulation value for the in-phase, prompt correlator arm

Q_PROMPT is the accumulation value for the quadrature, prompt correlator arm

I_LATE is the accumulation value for the in-phase, late correlator arm

Q_LATE is the accumulation value for the quadrature, late correlator arm

All the accumulator registers represent 16 bit 2's compliment numbers. These are overwritten on the tracking channels dump signal (~1ms period).

The **CARRIER_MEASUREMENT** register provides a 32 bit value representing the carrier cycle count and phase, at the TIC. Bits [31:10] are the carrier cycle count since the last TIC. Bits [9:0] are the carrier NCO phase (top 10 accumulator bits).

The **CODE_MEASUREMENT** register provides a 21 bit value representing the code half-chip number and code NCO phase at the TIC. Bits [20:10] are the code half chip number (0 to 2045). Bits [9:0] are the code NCO phase (top 10 accumulator bits).

The **EPOCH** register provides a count of the C/A code sequence latched on the TIC. There are two counters, the C/A code sequence counter and the 'bit' counter. The code sequence counter increments on the dump signal and counts from 0 to 19 (represented by bits [4:0]). The bit counter increments when the code sequence counter rolls over and counts 0 to 49 (represented by bits [10:5]).

The **EPOCH_CHECK** register provides the instantaneous epoch count. This register has the same structure as the epoch register, but is not latched on the TIC.

Writing to the **EPOCH_LOAD** register results in the epoch counters being set to the values loaded. The counters will then continue counting from those values. Bits [4:0] is the C/A code sequence counter (1ms period). Bits [10:5] represent the bit counter.

Status Registers

Bit 0 of **STATUS** is 1 if a TIC has occurred since the last read. Bit 1 is 1 if an accumulator interrupt has occurred since the last read. Cleared on read.

The **NEW_DATA** register provides accumulation status for each tracking channel. It is a bit map with tracking channel 0 represented by bit 0, channel 1 by bit 1 etc. Each bit is set to 1 when the corresponding channel accumulator registers have new data. This occurs on the channels 'dump' signal that is aligned to the beginning / end of the C/A code sequence. Note that each tracking channel has an independent dump cycle. Cleared on read.

The **TIC_COUNT** register provides the current value of the 24 bit TIC down counter.

The **ACCUM_COUNT** register provides the current value of the 24 bit ACCUM_INT down counter.

Control Registers

A write to the **RESET** register resets the entire tracking module including the time base functions.

The **PROG_TIC** register is used to set the TIC period. The value written to this register is used as a down counter.
TIC period = (p + 1) / clock frequency, where p is the value written to the register.

For a system with a 40MHz clock:
p = (40X10^6 X TIC_period) – 1
For a 100ms TIC, p = 0x3D08FF

The **PROG_ACCUM_INT** register is used to set the accumulator interrupt period. This period must be less than 1ms to ensure that accumulation data is not overwritten and lost.
ACCUM period = (p + 1) / clock frequency,
where p is the value written to the register.

For a system with a 40MHz clock:
p = (40X10^6 X ACCUM_period) – 1
For a 500us interrupt p = 0x4E1F

Resource Usage

The Namuru test bench system was built using Altera's 'Quartus' software on a Cyclone II FPGA part (EP2C35F), with 35,000 logic elements ('LEs'). The system includes a NiosII (II/f) soft-core processor with various peripherals (including JTAG and other serial comms) and extensive debug logic.

- Single tracking channel uses 575 LEs
- 12 channel tracking module uses 8890 LE's
- Full system uses 18201 LE's