NT1065

MMRFIC Technology Pvt. Ltd., Bangalore

June 13, 2020

VER 1

Ver.	Author(s)	Date	Comments
1.0	Sudhanshu Sharma	29/05/2020	Created

Table 1: Version History

Contents

1 Introduction **3**

2 NT1065 **3**

3 Features of NT1065 **3**

4 Applications **4**

5 NT1065.h **4**

6 NT1065.c **4**

7 NTCONFIG.c **5**

8 Application notes **7**

8.1 REFERENCE FREQUENCY (TCXO) CONFIGURATION AND START UP PROCEDURE 7

8.2 PLL A/ PLL B RECONFIGURATION 8

8.3 SINGLE LO SOURCE CONFIGURATION 8

8.4 2-BIT ADC CONFIGURATION 8

8.5 CLK FREQUENCY CONFIGURATION 9

8.6 TEMPERATURE MEASUREMENT PROCEDURE 9

1 Introduction

This document contains the documnentation regarding NT1065. It explains the features of NT1065 along with the content of the files like NT1065.c, NT1065.h, NT1065cofig.h.

2 NT1065

NT1065 is a four-channel RF Front-End IC for the reception of Global Navigation Satellite System (GNSS) signals (GPS, GLONASS, Galileo, BeiDou, NavIC, QZSS) and also signals of satellite-based augmentation systems like OmniSTAR at all frequency bands in various combinations: L1, L2, L3, L5, E1, E5a, E5b, E6, B1, B2, B3. Galileo E5 band as well as BeiDou B1, B2, B3 (phase 3) band can be obtained as entire signal with two channels fed by the same LO and then restored in digital domain to true complex data. As a benefit one can discover wide possibilities of improving the positioning accuracy down to centimeter range without taking RTK technique. Each setting, including output signal frequency bandwidth, AGC options, mirror channel suppression option, etc., can be set for every channel individually. NT1065 includes two fully independent frequency synthesizers. Channel 1 and channel 2 are supplied with LO signal generated in PLL "A" while PLL "B" is assigned for channels 3 and 4. For specific applications there is an option to feed all four channels with single LO source from PLL "A". This powerful toolkit is accompanied with very simple and easy-to-use register map. All the functionality allows application of NT1065 in high precision GNSS based positioning, goniometric, driverless car systems and related branches

3 Features of NT1065

- Single conversion super heterodyne receiver
- Four independent configurable channels, each includes preamplifier, image rejection mixer, IF filter, IFA, 2-bit ADC
- Signal bandwidth up to 31MHz supports GNSS high precision codes such as P-code in GPS or wideband E5 Galileo
- Dual adoptable AGC system (RF + IF) or programmable gain
- High dynamic range with 1dB compression point more than -30dBm
- Analog differential output with two options of voltage swing 0.2/0.47Vp-p and 0.4/0.98Vp-p (sine wave/noise) or 2-bit ADC digital output data
- Two independent fully integrated synthesizers with flexible LO and CLK frequencies selection ("A" and "B")
- Embedded temperature sensor
- SPI interface with easy-to-use register map

- Individual status indicators of main subsystems (available in SPI registers) and cumulative status indicator (AOK, available both as a separate pin and in SPI registers)
- 10x10mm QFN88 package

4 Applications

- GNSS based positioning systems
- GNSS based goniometric systems
- In-vehicle navigation systems
- GNSS based driverless car systems
- Professional drones

5 NT1065.h

This sections describe how the .h file is written. It has all the structure defined for all 48 registers and their enum.


```
struct {
    unsigned short R0_VAL;

}NT_REG_STRUCT;

typedef enum {
    R0_ADDR = 0,

}NT_ADDR_STRUCT;

struct {
    unsigned short defaultval;
    unsigned short val;
}r0;
```

6 NT1065.c

This sections describe how the .c file is written. It has all the register value defined and also the corresponding initData.

```
    r0.defaultval=33;
    r0.val = r0.defaultval;
     NT_REG_STRUCT.R0_VAL = r0.val;
```

7 NTCONFIG.c

This sections describe how the .c file is written. It has all the configuration done to use NT1065 for project.

```
 // SETTING TO BE PERFORMED FOR FREQUENCY = 16.368MHz. PLL A setting for NT1065

initData.valueofREG67 = 0xCD;
initData.valueofREG68 = 0x8A;
PPLSel = 1;                          //REG2[1-0] 0->standby,
        //1->PLL A, 2->PLL B,3->Active
        initData.PPLSel = PPLSel;   //REG12
targetFreqMHz = 16368000;

        // CALCULATING VALUE OF R AND N

R = 1;                    // to assign the value of R(1,15)
N = 97;                 // Value given by gana
     //N = targetFreqMHz * R/TCXOfreq;   // value of N(48,511)

        // PLL A write REG41 and for PLL B  REG45
PllBand = 1 ;                  //REG41(1) 0->L2/L3/L5, 1->L1
initData.PllBand = PllBand;


// SPLITTING THE VALUE OF N

N2 = N&256; // '100000000' is 256
N2 = N2>>8;
N1 = N&255; // '11111111' is 255


initData.N1 = N1;         // REG42(7-0) for PLL A and REG46(7-0) for PLL B
initData.N2 = N2;         // REG43(7) for PLL A and REG47(7])for PLL B
initData.R = R;            // REG43(6-3) for PLL A and REG47(6-3) for PLL B

     // DELAY of 1ms to be given to lock PLL


        // 7.3 SINGLE LO SOURCE CONFIGURATION

signalLOConfigA = 0;      // REG3[0] feed all mixers from PLL "A"
initData.signalLOConfigA =signalLOConfigA;
signalLOConfigB = 0; // REG45[0] turning off PLL"B"
initData.signalLOConfigB = signalLOConfigB;
```

```
         //7.4 RF AGC CONFIGURATION

rfAgc = rcG12p00 ;       // REG17(7-4) see enum for REG17 to assign value
initData.rfAgc =rfAgc;
LPFCali = passband15p69;   //REG14(6-0) for channel "A" then REG21,REG28,REG35
initData.LPFCali = LPFCali;


        //7.7 "0" analog differential  "1" 2-bit ADC output R15

ADCoutput = 1;

// REG15(0)
X  = 1 ;           //User has to give value
if (ADCoutput==1)

switch (X)
case 1: Adctype = 0; //REG19
break;
case 2: Adctype = 1;
break;
case 3: Adctype = 2;
break;
case 4: Adctype = 3;
break;



// 7.8 Clock freq calculation

C = 8;        //Given by gana sir can be from value 8 - 31 with step 1
freqclk = targetFreqMHz/2*C;
PPLSel = 0; // REG12      // select PPL "A" - 0, PLL "B" - 1
initData.C=C; //REG11
initData.PPLSel = PPLSel; //REG12
initData.Adctype=Adctype; //REG19
initData.ADCoutput=ADCoutput; // REG15


// 7.10

TempMode = 1; // 1 for continous and 0 for single
initData.TempMode = TempMode; // REG5 D 0
```

8 Application notes

8.1 REFERENCE FREQUENCY (TCXO) CONFIGURATION AND START UP PROCEDURE

After power up NT1065 assumes feeding with 10MHz TCXO signal and wakes up in the active mode. PLLs are supposed to be locked after 1 ms and generally chip is ready for operation. During next 15 ms LPF calibration procedure is running in background mode and has no influence on channel filters. After completion a cut-off frequency correction code is applied to all channels automatically and NT1065 has following configuration:

- PLL A is set to L1 band and feeds channel1 and channel2 with LO = 1590 MHz
- PLL B is set to L2/L3/L5 band and feeds channel3 and channel4 with LO = 1235 MHz
- Channel1 down converts low side band (i.e. L1 GPS/Galileo/BeiDou/QZSS)
- Channel2 down converts high side band (i.e. L1 GLONASS)
- Channel3 down converts high side band (i.e. L2 GLONASS)
- Channel4 down converts low side band (i.e. L2 GPS/QZSS)
- All channels are set to analog differential output data interface, RF GC system in manual mode at max gain, IF GC system in auto mode PLL A and PLL B tuning systems were executed
- LPF auto calibration system was executed
- 53 MHz CLK of LVDS type is pushed out IF non 10MHz TCXO is used, some actions should be performed in order to make NT1065 perform properly.

16.368 MHz TCXO:

- upload following values to Reg67 and Reg68 (thereafter avoid Reg3 D[1] changing, which will reset TCXO configuration to predefined state):
  Reg67 xCD
  Reg68 x8A
- perform PLL A and PLL B (if intended to use) reconfiguration according to section 7.2 to get desired LO frequency
- execute PLL A and PLL B (if intended to use) auto tuning procedure - Reg43 D[0] and Reg47 D[0] correspondingly
- execute LPF auto-calibration system - Reg4 D[0].

8.2 PLL A/ PLL B RECONFIGURATION

In order to reconfigure PLL following procedure is recommended:

- set Reg41/Reg45 D[1] to desired frequency band
- using the formula:

$$F_{LO} = \frac{NF_{TCXO}}{R} \tag{1}$$

  choose N and R
- write N value to Reg42/Reg46 D[7-0] + Reg43/Reg47 D[7]
- write R value to Reg43/Reg47 D[6-3]
- execute tuning procedure - Reg43/Reg47 D[0]

PLLs need 1 ms to be locked.

8.3 SINGLE LO SOURCE CONFIGURATION

In order to switch to single LO mode following actions are to perform:

- set Reg3 D[0] to '0' to feed all mixers from PLL A
- turn off PLL B by setting Reg45 D[0] to '0'
- reconfigure PLL A according to desired frequency plan using Reg41-44 if needed

8.4 2-BIT ADC CONFIGURATION

After power up NT1065 is preconfigured to analog differential output data interface. However, there is an option to set up 2-bit ADC outputs in Reg15 D[0] for Channel 1 / Reg22 D[0] for Channel2 / Reg29 D[0] for Channel3 / Reg36 D[0] for Channel 4. 2-bit ADCs are able to operate in one of three modes:

- clocked by rising edge;
- clocked by falling edge;
- asynchronous.

These modes can be set up in Reg19 D[3-2] for Channel1 / Reg26 D[3-2] for Channel2 / Reg33 D[3-2] for Channel3 / Reg40 D[3-2] for Channel4. For ADCs sampling frequency information, please, refer to subsection 7.7. In "asynchronous" mode 2-bit ADCs act as voltage level comparators so no any clocking applied. For example, this mode may be use full if several NT1065s should operate simultaneously pushing out digitized data that can be synchronized with single clock on correlator and processor side

8.5 CLK FREQUENCY CONFIGURATION

CLK signal is intended for clocking all 2-bit ADCs as well as clocking external correlator engine. It is generated from LO frequency either from PLL "A" or PLL "B" according to the formula:

$$F_{CLK} = \frac{F_{LO}}{2 * C} \tag{2}$$

CLK sources and frequency can be customized by procedure

- choose CLK source by setting appropriate value to Reg12 D[5]
- write C value to Reg11 D[4-0]

8.6 TEMPERATURE MEASUREMENT PROCEDURE

Two modes of temperature modes are available: single and continuous (Reg5 D[1]). In single mode the measurement is done once upon request to Reg5 D[0] by setting '1' and result will be stored in Reg7 D[1-0] + Reg8 D[7-0] after procedure is finished (auto reset to '0' in Reg5 D[0] indicates this) until next execution. One temperature measurement procedure time is up to 17 ms. To enter in continuous mode set Reg5 D[1] to '1' first then execute with Reg5 D[0]. In this case embedded temperature sensor periodically runs the measurement procedure and only the latest result is stored in Reg7 D[1-0] + Reg8 D[7-0]. In order to stop continuous execution Reg5 D[1] should be set to '0'.