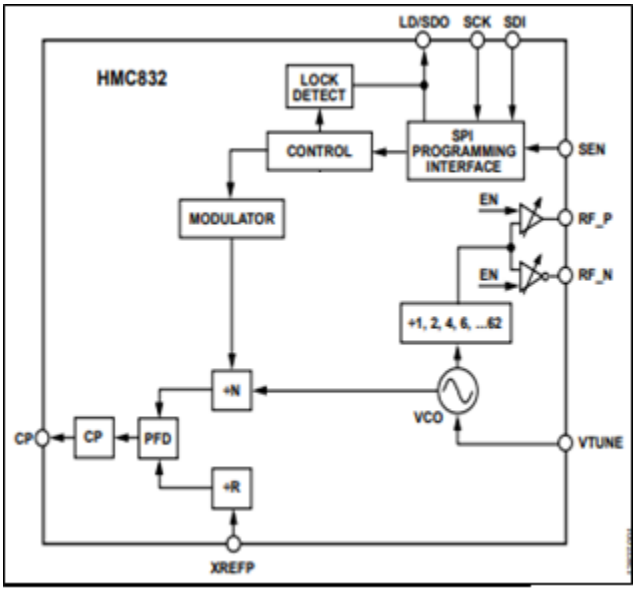
Research Project

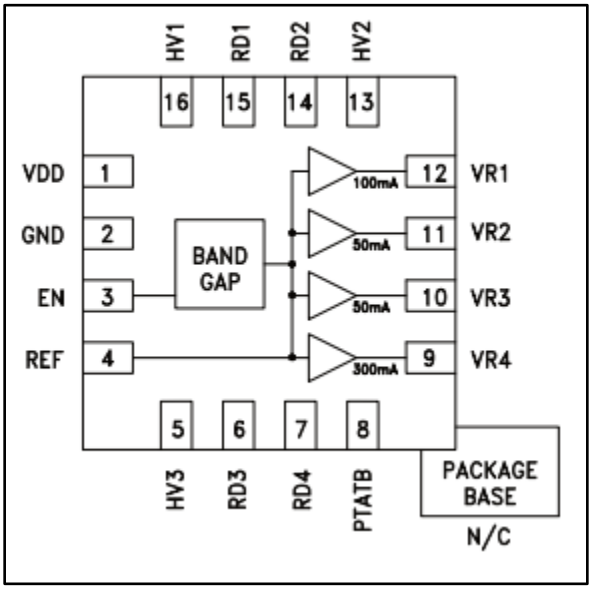
Muhammad Shamaas 18100217

HMC832LP6GE Fractional N PLL with Integrated VCO

HMC832 Functional Block Diagram



HMC1060LP3E Voltage Regulator Functional Block Diagram



HMC832 Pin Voltages:

Pins 2, 5, 6, 8, 9, 11 to 14, 18 to 22, 24, 26, 34, 37, 38: Not Connected.

Pins 1, 3, 7, 10, 16, 17, 25, 27, 35, 36, 39: 3.3V (Min 3.1 V – Max 3.5 V).

Pin 4: Charge Pump Output. Connected to Loop Filter.

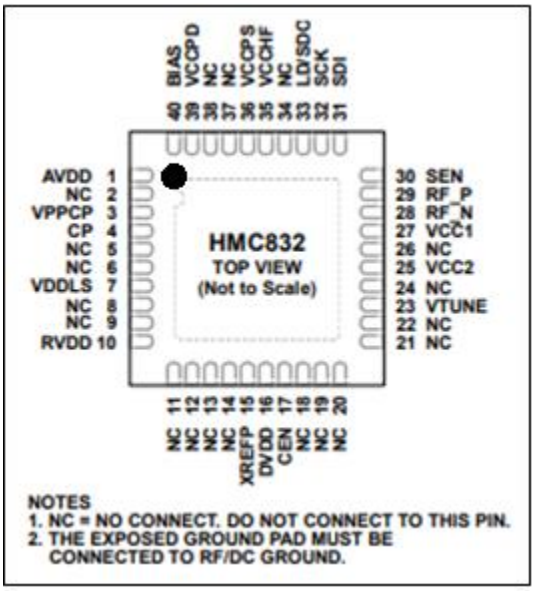
Pin 23: Tune Voltage Input. Connected to Loop Filter.

Pin 15: Reference Oscillator Input.

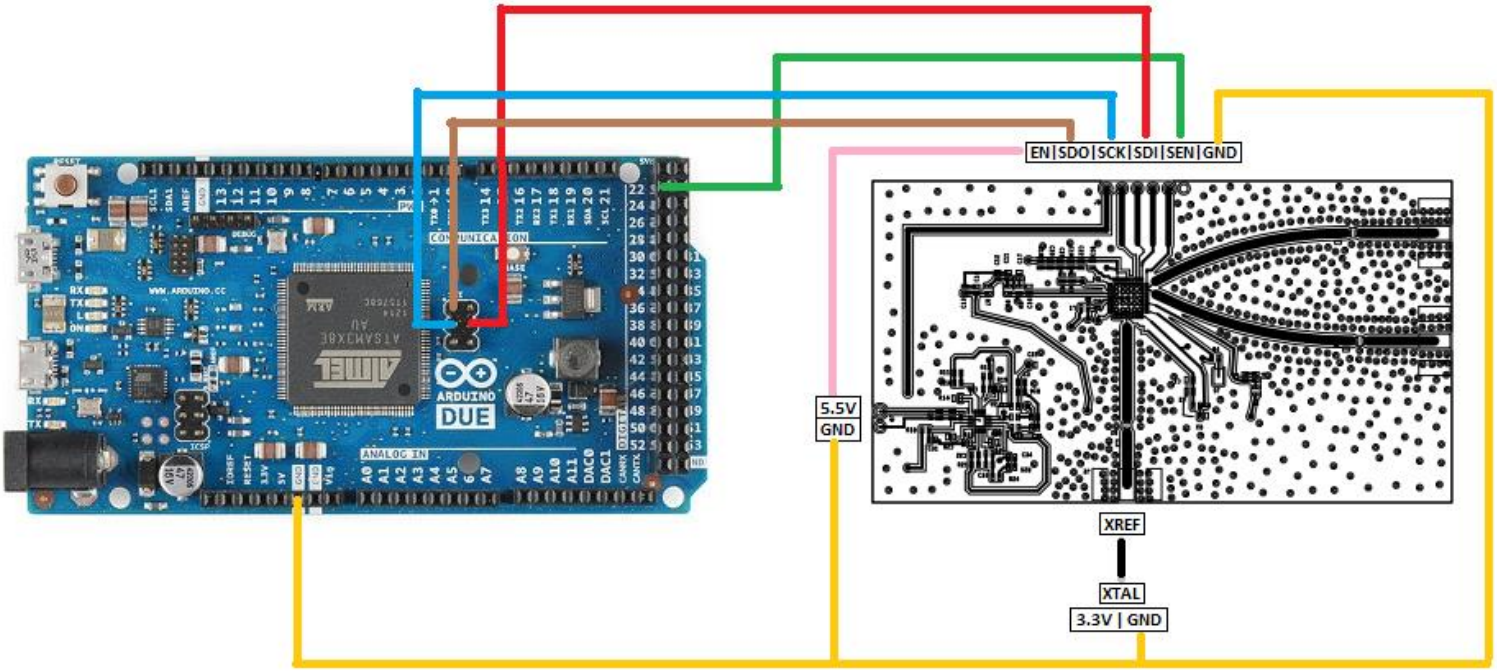
Pin 28-29: VCO Outputs.

Pin 30-33: SPI Interface (pin 30: SEN, pin 31: SDI, pin 32: SCK, pin 33: LD/SDO)

Pin 40: External Bypass Decoupling for Precision Bias Circuits.



Connections



As visible in the PCB print, the path leading from XREF to the HMC832 IC has a VIA which interrupts that path and grounds the XREF input. To correct this problem, the via was scratched using a blade and it was checked that the XREF reached the metal pin on the IC. The VCO subsystem cannot be programmed without XREF.

PLL Subsystem Register map

	Variables							
0x00	CHIP_ID							
0x00	Read Address	Soft Reset	Not Defined					
0x01	RST_CHIPEN_PIN_SELECT	RST_CHIPEN_FROM_SPI	Reserved					
0x02	RDIV							
0x03	INTG_REG							
0x04	FRAC							
0x05	VCO_ID	VCO_REGADDR	VCO_DATA					
0x06	Seed	Reserved	FRAC_BYPASS	Initialization	SD enable	Reserved	Automatic clock configuration	Reserved
0x07	LKD_WINCNT_MAX	Enable internal lock detect	Reserved	Lock detect window type	LD digital window duration	LD digital timer frequency control	Reserved	Automatic relock: one try
0x08	Reserved	GPO_PAD_EN	Reserved	VCO buffer and pre-scaler bias enable	Reserved	High frequency reference	Reserved	
0x09	CP DN gain	CP UP gain	Offset magnitude	Offset up enable	Offset DN enable	Reserved		
0x0A	VTUNE resolution	Reserved	Force curve	Auto calibration disable	No VSPI trigger	FSM/VSPI clock select	Reserved	
0x0B	PD_DEL_SEL	Reserved	PD_UP_EN	PD_DN_EN	CSP mode	Force CP up	Force CP DN	Reserved
0x0C	Number of Channels per fPD							
0x0F	GPO_SELECT	GPO test data	Prevent automux SDO	LDO driver always on	Disable PFET	Disable NFET		
0x10	VCO switch setting	Auto calibration busy						
0x11	SAR error magnitude counts	SAR error sign						
0x12	GPO	Lock detect						
0x13	Reserved							

VCO Subsystem Register map

	Variables										
0x00	CAL	CAPS									
0x01	Master enable VCO subsystem	VCO enable	PLL buffer enable	Input/output master enable	Reserved	Output stage enable	Reserved	Reserved			
0x02	RF divide ratio	Reserved									
0x03	Programmable performance mode	RF_N output enable					RF_P output enable	Reserved	Return loss	Reserved	Mute mode
0x04	Initialization										
0x05	Reserved										
0x06	Reserved										
0x07	Output stage gain control								Initialization		Reserved

HMC832 Technical Details:

1. **Bandwidth:** 25 – 3000 MHz
2. **Input Supply:** 3.3 V (Min 3.1 V – Max 3.5 V).
3. **Absolute Maximum Pin Ratings:** -0.3V to +3.6V.
4. **Output Gain:** 0-11 dB
5. **Integrated Phase detector:** Maximum 100 MHz
6. **Delta Sigma Modulator.**
7. **Built in Self-Test and Lock Detect.**
8. **Cycle Slip Prevention.**
9. **In-Band Phase Noise Floor:** -110 dBc/Hz.
10. **Fractional Figure of Merit:** -226 dBc/Hz.
11. **Single ended or differential output.**
12. **Power on Reset:** All Registers are programmed to default values at power up. It takes 250us. By default, Fractional Mode is enabled with Auto-calibration and the Outputs are not enabled. However, if Power on Reset is successful, a faint peak at 2310MHz is visible on the frequency spectrum and HMC832 enters Ready mode where it has to select SPI mode, based on whether rising edge occurs first on SCK or SEN pin. The default variables are given below. Their descriptions are included in code.

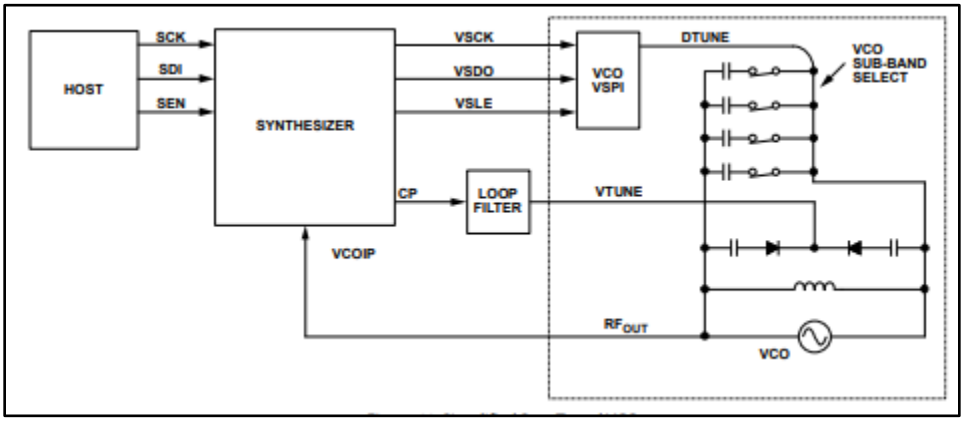
PLL REGISTERS DEFAULT VARIABLES
RST_CHIPEN_PIN_SELECT=0;
RST_CHIPEN_FROM_SPI=1;
RDIV=1;
INT_REG=25;
FRAC=0;
Seed=2;
FRAC_BYPASS=0;
SD_ENABLE=1;
Automatic_Clock_Configuration=1;
LKD_WINCNT_MAX=5;
Enable_Internal_LockDetect=1;
LD_Window_Type=1;
LD_Digital_Window_Duration=2;
LD_Digital_Timer_Frequency_Control=0;
Automatic_Relock=0;
GPO_PAD_EN=1;
VCO_Buffer_and_Prescalar_Bias_Enable=1;
High_Frequency_Reference=0;
CP_DN_gain=100;
CP_UP_gain=100;
Offset_Magnitude=0;
Offset_UP_enable=0;
Offset_DN_enable=1;
VTUNE_Resolution=5;
Force_Curve=0;
AutoCalibration_Disabled=0;
No_VSPI_Trigger=0;
FSM_VSPI_Clock_Select=1;
PD_DEL_SEL=1;
PD_UP_EN=1;
PD_DN_EN=1;
CSP_Mode=0;
Force_CP_UP=0;
Force_CP_DN=0;
Number_of_channels_per_fPD=0;
GPO_SELECT=1;
GPO_Test_Data=0;
Prevent_Automux_SDO=0;
LDO_driver_always_on=0;
Disable_PFET=0;
Disable_NFET=0;

VCO REGISTERS DEFAULT VARIABLES
CAL=0;
CAPS =16;
Master_Enable_VCO=1;
VCO_Enable=1;
PLL_Buffer_Enable=1;
I_O_master_Enable=1;
Output_Stage_Enable=1;
RD_divide_ratio=1;
Programmable_Performance_Mode=2;
RF_N_Output_Enable=0;
RF_P_ Output_Enable=0;
Return_loss=0;
Mute_mode=1;
Output_Stage_gain=1;

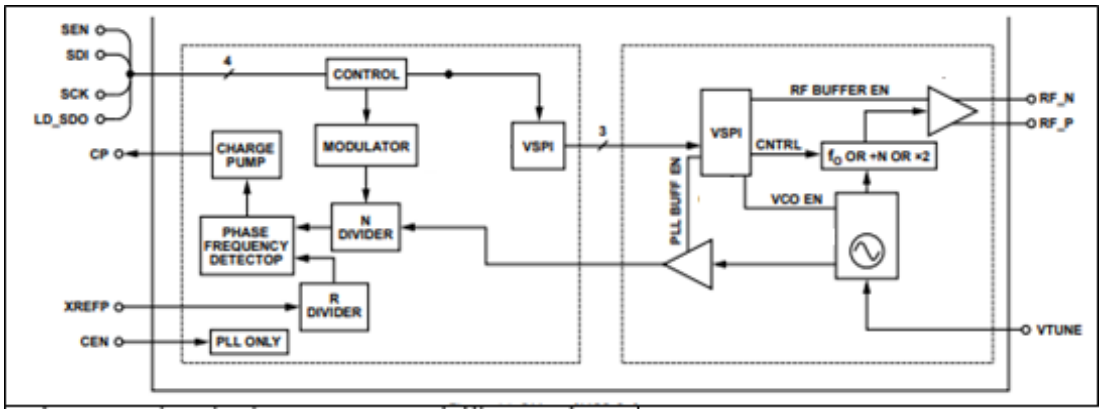
13. **GPO Pin:** Configurable General-Purpose Output Pin Lock Detect/ Serial Data Out (LD/ SDO pin).

Bit	Type	Name	Width	Default	Description
4:0	R/W	GPO_SELECT	5	1d	The signal selected here is an output to the SDO pin when the SDO pin is enable via Register 0x08[5] 0: data from Register 0x0F[5] 1: lock detect output 2: lock detect trigger 3: lock detect window output 4: ring oscillator test 5: pull-up hard from CSP 6: pull-down hard from CSP 7: reserved 8: reference buffer output 9: reference divider output 10: VCO divider output 11: modulator clock from VCO divider 12: auxiliary clock 13: auxiliary SPI clock 14: auxiliary SPI enable 15: auxiliary SPI data output 16: PD down 17: PD up 18: SD3 clock delay 19: SD3 core clock 20: autostrobe integer write 21: autostrobe fractional write 22: autostrobe auxiliary SPI 23: SPI latch enable 24: VCO divider sync reset 25: seed load strobe 26 to 29: not used 30: SPI output buffer enable 31: soft reset, $\overline{RST}$
5	R/W	GPO test data	1	0	1: GPO test data

14. **VCO:** 256 sub-bands and range 1500MHz – 3000MHz. Either VTUNE or CAPS (VCO_REG_00) can be used to select VCO sub-band or change VCO frequency.

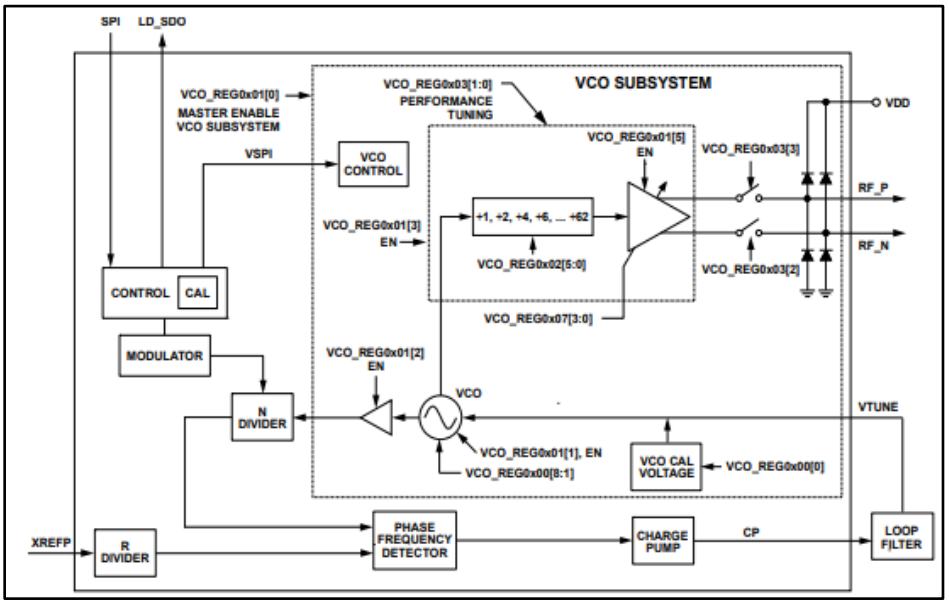


15. **Two Subsystems:** PLL subsystem and VCO subsystem.

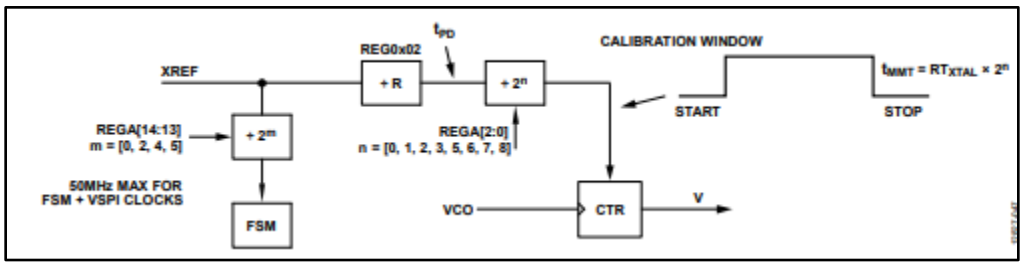


PLL subsystem: Programmable through external SPI Clock. It is enabled through CEN pin. PLL Subsystem Registers can be written and read back as well.

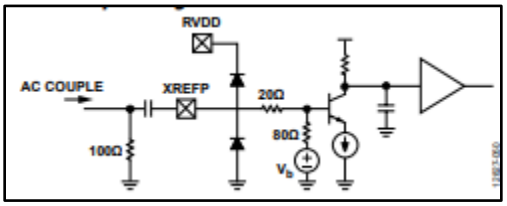
VCO subsystem: Indirectly addressed through PLL_REG_05. VCO Subsystem Registers can be written but not read back.



VCO Subsystem is programmed through FSM/ VSPI Clock (Derived from XREF: External Reference Source Frequency). If External Reference Crystal power is lower than threshold, VCO subsystem will not be programmed correctly.

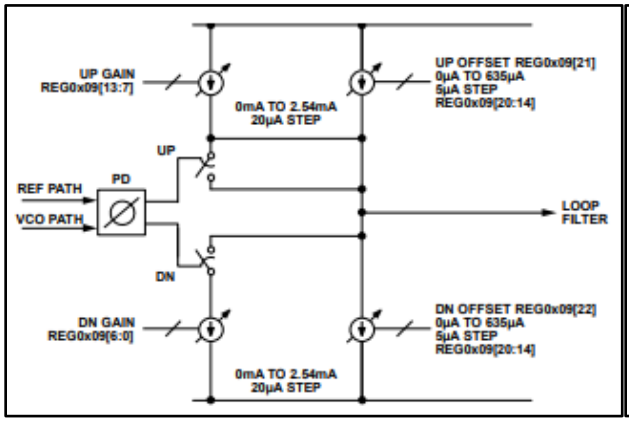


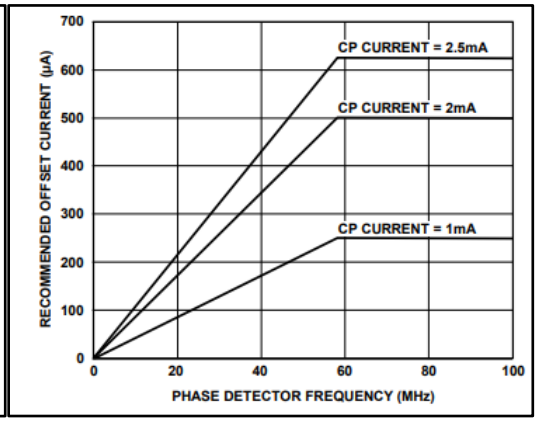
16. **Reference Input:** Maximum 350 MHz at +12 dBm (AC Coupled). Recommended frequency, shape and power are given in the table below.



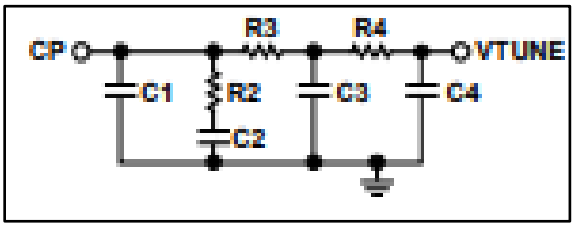
Reference Input Frequency (MHz)	Square Input			Sinusoidal Input		
	Slew > 0.5 V/ns	Recommended Swing (V p-p)			Recommended Power Range (dBm)	
	Recommended	Minimum	Maximum	Recommended	Minimum	Maximum
<10	Yes	0.6	2.5	No	No	No
10	Yes	0.6	2.5	No	No	No
25	Yes	0.6	2.5	Okay	8	15
50	Yes	0.6	2.5	Yes	6	15
100	Yes	0.6	2.5	Yes	5	15
150	Okay	0.9	2.5	Yes	4	12
200	Okay	1.2	2.5	Yes	3	8

17. **Charge Pump:** Connected between output of Phase Detector (PD) and input of Loop Filter. Both Up and Down gains (Typically 2-2.5 mA) must be equal. The Up and Down Offsets must be non-zero for Fractional Mode only. Offset value depends on Phase Detector Frequency (XREF/R).





18. **Loop Filter:** Type 2 Loop Filter was connected between CP and VTUNE pins.



Loop Filter Type	Loop Filter BW (kHz)	Loop Filter Phase Margin	C1 (pF)	C2 (nF)	C3 (pF)	C4 (pF)	R2 (Ω)	R3 (Ω)	R4 (Ω)	Loop Filter Design
Type 1[1]	127	61°	390	10	82	82	750	300	300	See Figure 52
Type 2[2]	75	61°	270	27	200	390	430	390	390	
Type 3[3]	214	71°	56	1.8	NA	NA	2200	0	0	

[1] Loop Filter Type 1 is for best integrated phase noise. Loop filter bandwidth is designed for 50 MHz PD frequency, CP = 1.6 mA at 2.2 GHz output in fractional mode.
[2] Loop Filter Type 2 is suggested to use for best far out phase noise. Loop filter BW is designed for 50 MHz PD frequency, CP = 1.6 mA at 2.2 GHz output in fractional mode.
[3] Loop Filter Type 3 is suggested to use for best integrated phase noise at integer mode. Loop filter bandwidth is designed for 50 MHz PD frequency, CP = 2.5 mA at 3 GHz output in integer mode.

Modes of Operation:

1. **Output Mute and Power Down Function:** VCO and PLL must be shut down separately.

2. **Fundamental Frequency Mode:** 1500 MHz – 3000 MHz
   The VCO sub-band is chosen using VCO_REG_00 (CAPS).

3. **Integer mode:** 25 MHz – 3000 MHz
   Both Fractional Modulator and Delta-Sigma Modulator must be disabled.
   Output Frequency Divider Ratios of 1/2/4/6/.../62 are available.
   N_{INT} Divider may be programmed to any integer from 16 to 524287.
   $F_{out} = \frac{(\frac{XREF}{R})(N_{INT})}{Output\ Divider}$.
   Charge Pump offset must be zero.
   Typically, the Charge Pump Up and Down gains must be equal and non-zero.
   If Auto calibration is disabled, the VCO sub-band is chosen using VCO_REG_00 (CAPS).
   If Auto calibration is enabled, the VCO sub-band is chosen automatically using VTUNE by internal Finite State Machine after the PLL Registers 0x03 or 0x04 are written.

4. **Fractional Mode:** 25 MHz – 3000 MHz
   Both Fractional Modulator and Delta-Sigma Modulator must be enabled. Output Frequency Divider Ratios of 1/2/4/6/.../62 are available.
   N_{INT} Divider may be programmed to any integer from 16 to 524284.
   Fractional Divider value N_{FRAC} can be set from $(\frac{1}{2^{24}})$ to $(\frac{2^{24}-1}{2^{24}})$.
   $F_{out} = \frac{(\frac{XREF}{R})(N_{INT}+N_{FRAC})}{Output\ Divider}$.
   Both Charge Pump gain and Offset may be non-zero.
   If Auto calibration is disabled, the VCO sub-band is chosen using VCO_REG_00 (CAPS).
   If Auto calibration is enabled, the VCO sub-band is chosen automatically using VTUNE by internal Finite State Machine after the PLL Registers 0x03 or 0x04 are written.

5. **Exact Frequency Mode:** 0 Hz error can be achieved for special frequencies in Fractional and Integer Frequency Mode.

6. **Auto calibration Function:** The VCO sub-band is selected automatically using VTUNE by the internal Finite Sate Machine after the PLL Registers 0x03 or 0x04 are written.

SPI Modes:

1. **HMC Mode:** First Rising Edge must occur on SEN pin before SCK pin after Power On Reset. My Code and HMC832 Evaluation Software use HMC Mode for SPI Write.

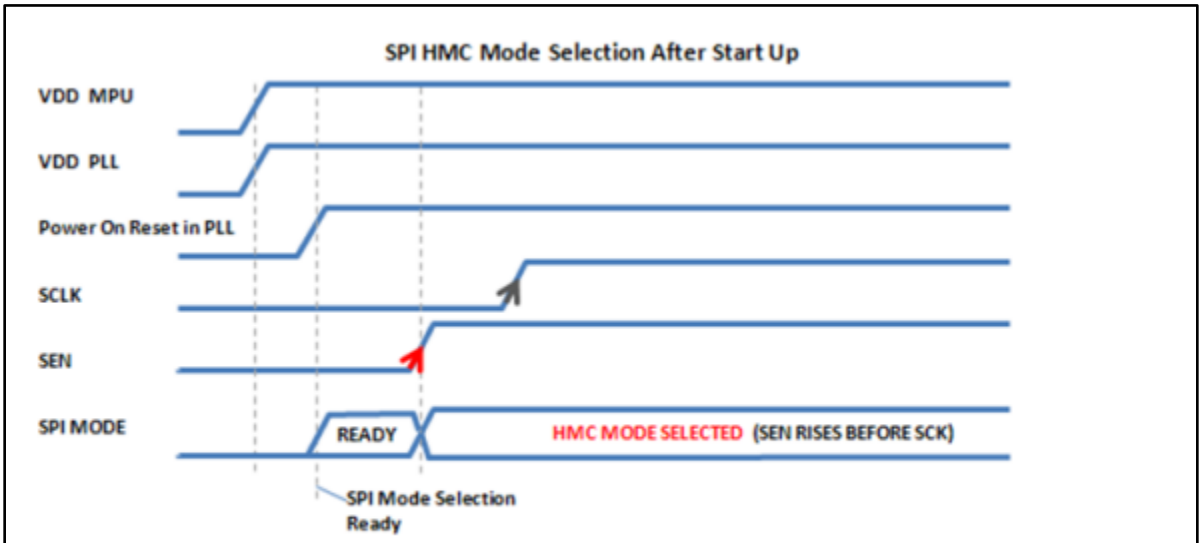
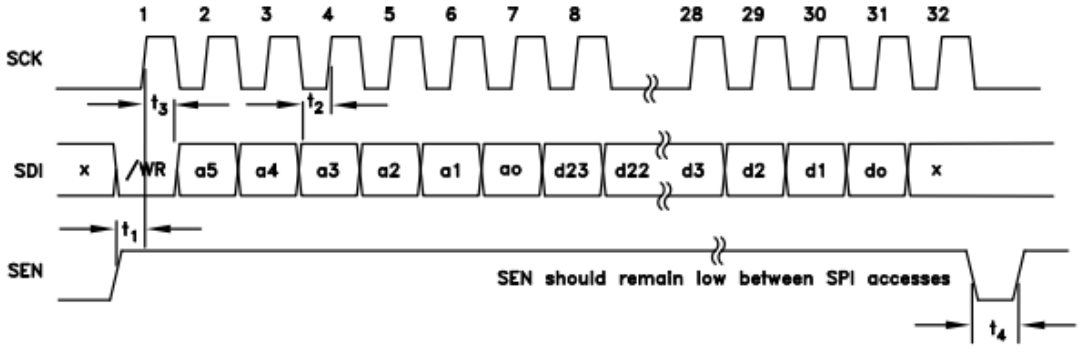


Figure 3 SPI HMC Mode Selection, MPU Guaranteed Low on Start Up

HMC Mode Write Timing Diagram.

a. The Master (host) both asserts SEN (Serial Port Enable) and clears SDI to indicate a WRITE cycle, followed by a rising edge of SCK.
b. The slave (synthesizer) reads SDI on the 1st rising edge of SCK after SEN. SDI low indicates a Write cycle (/WR).
c. Host places the six address bits on the next six falling edges of SCK, MSB first.
d. Slave shifts the address bits in the next six rising edges of SCK (2-7).
e. Host places the 24 data bits on the next 24 falling edges of SCK, MSB first.
f. Slave shifts the data bits on the next 24 rising edges of SCK (8-31).
g. The data is registered into the chip on the 32nd rising edge of SCK.
h. SEN is cleared after a minimum delay of t_5. This completes the write cycle.



2. **Open Mode:** First Rising Edge must occur on SCK pin before SEN pin after Power On Reset.

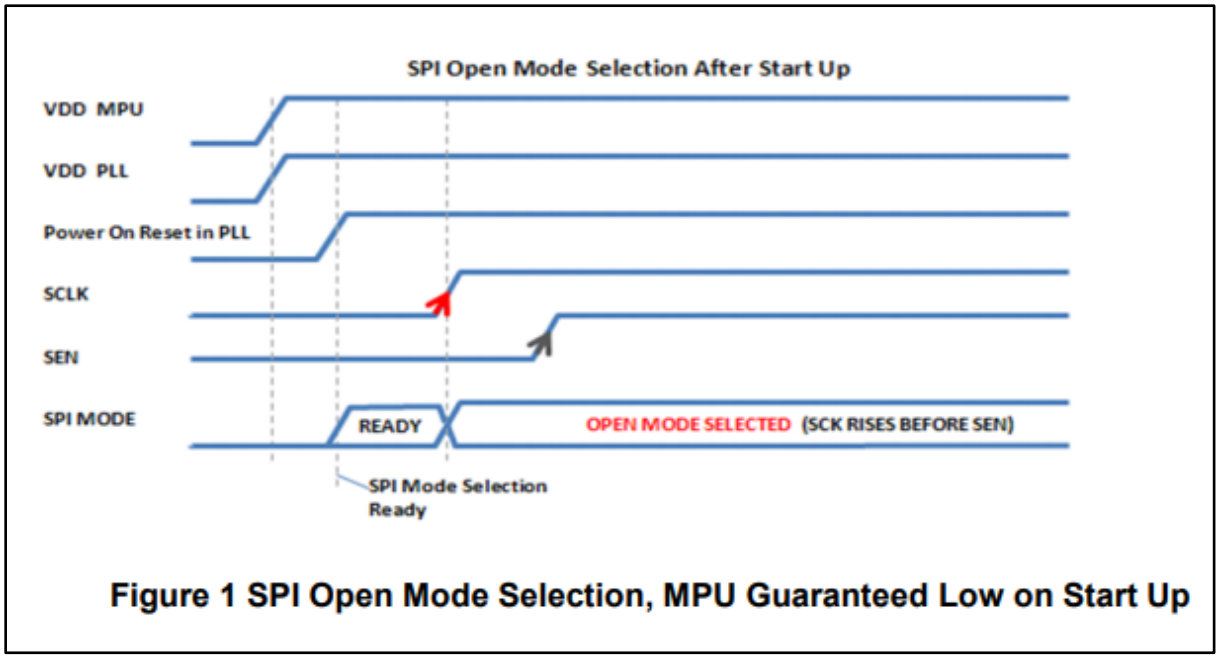


Figure 1 SPI Open Mode Selection, MPU Guaranteed Low on Start Up

HMC Mode Write Timing Diagram.

a. The Master (host) places 24-bit data, d23:d0, MSB first, on SDI on the first 24 falling edges of SCLK.
b. the slave (PLL with Integrated VCO) shifts in data on SDI on the first 24 rising edges of SCLK
c. Master places 5-bit register address to be written to, r4:r0, MSB first, on the next 5 falling edges of SCLK (25-29)
d. Slave shifts the register bits on the next 5 rising edges of SCLK (25-29).
e. Master places 3-bit chip address, a2:a0, MSB first, on the next 3 falling edges of SCLK (30-32). Hittite reserves chip address a2:a0 = 000 for all RF PLL with Integrated VCOs.
f. Slave shifts the chip address bits on the next 3 rising edges of SCLK (30-32).
g. Master asserts SEN after the 32nd rising edge of SCLK.
h. Slave registers the SDI data on the rising edge of SEN.

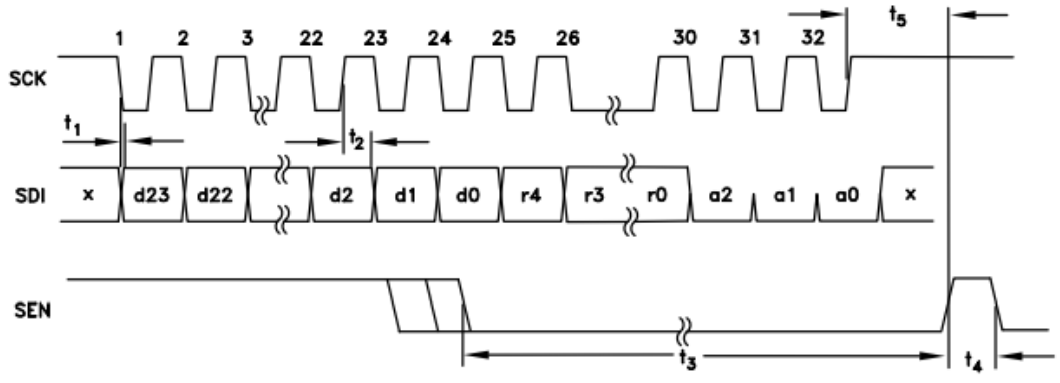
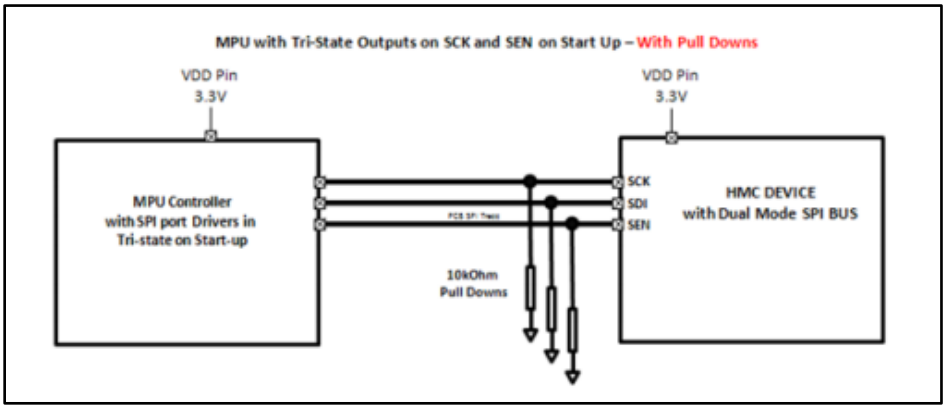


Figure 26. Open Mode - Serial Port Timing Diagram - WRITE

Instructions:

1. Power off the Arduino and crystal source before powering on the PLL. Power on the PLL IC followed by the crystal source and then the Arduino. Automatic Power on Reset takes 250us.
2. Registers have default values after Automatic Power on Reset. All Registers need to be programmed after power on Reset.
3. Write PLL_REG_00 = 0x000000 first upon power up.
4. At power-up, it is required that both SEN and SCK lines are initially held low. If the first rising edge occurs on the SCK line before any rising edges occur on the SEN Pin, SPI Open Mode is selected. If the first rising edge occurs on the SEN line before any rising edges occur on the SCK Pin, SPI HMC Mode is selected. My Code only works if HMC Mode is chosen.



5. For write to PLL subsystem, Register Address and Data must be written.
   The data is broken down into two fields.

6-bit Address	24-bit Data

6. For write to VCO subsystem, PLL_REG_05 must be written.
   The data is broken down into three fields. Bits [2:0]: 3-bit VCO_ID (always 0b000), Bits [6:3]: 4-bit VCO_REGADDR, Bits [15:7]: 9-bit VCO_DATA.

6-bit Address (0x5)	0x00 (eight 0-bits)	9-bit VCO Data	4-bit VCO_REG Address	3-bit VCO_ID

   The VCO subsystem is written at a clock derived from the input reference clock (FSM or VSPI clock = XREF divided by 1/4/16/32). No other SPI activity must take place while VCO subsystem is being written (32 clock cycles of VSPI clock).
7. The last write must be to PLL_REG_03 in Integer mode; and PLL_REG_03 followed by PLL_REG_04 in Fractional mode. This applies when Auto calibration is enabled or disabled.
8. The VCO sub-band can be set using the CAPS variable. There are 256 sub-bands for the VCO range 1500MHz to 3000MHz.
9. For Auto calibration mode, write all PLL registers, followed by all VCO registers. After the last write, write VCO_REG_00 = 0x00, followed by PLL_REG_03 if using Integer Mode; or PLL_REG_03 and then PLL_REG_04 if using Fractional mode.
10. If the output does not meet required accuracy, use the HMC832 Evaluation Kit Software to find appropriate register values for a better frequency match and then program those register values.

Observations:

Power Up Voltages and Currents

1. DC Voltage and Current for HMC1060 Voltage Regulator = 5.5V, 200 mA
2. DC Voltage and Current for Reference Crystal IC = 3.3V, 150 mA

Frequency Locked at 2GHz in Integer Mode

3. N-Channel Output Power = +1.5dBm
4. P-Channel Output Power = -8.9dBm
5. DC Voltage and Current for HMC1060 Voltage Regulator = 5.5V, 202.2 mA
6. DC Voltage and Current for Reference Crystal IC = 3.3V, 162.1 mA

Register Settings for three LO Banks.

Register	GPS (1621.84 MHz)	Beidou (1607.196 MHz)	GLONASS (1649 MHz)
0x00	A7975	A7975	A7975
0x01	2	2	2
0x02	1	1	1
0x03	51	50	52
0x04	178D50	5C1BDA	733333
0x06	F4A	F4A	F4A
0x07	14D	14D	14D
0x08	C1BEFF	C1BEFF	C1BEFF
0x09	3FFEFD	3FFEFD	3FFEFD
0x0A	2046	2046	2046
0x0B	F8061	F8061	F8061
0x0C	0	0	0
0x0F	81	81	81
0x10	EA	EB	E8
0x11	80006	80008	13
0x12	0	0	0
0x13	1259	1259	1259

Variable	GPS	Beidou	GLONASS
RST_CHIPEN_PIN_SELECT	0	0	0
RST_CHIPEN_FROM_SPI	1	1	1
Reserved	0	0	0
RDIV	1	1	1
INT_REG	81	80	82
FRAC	1543504	6036442	7549747
Seed	2	2	2
Reserved	18	18	18
FRAC_BYPASS	0	0	0
Initialization	7	7	7
SD_ENABLE	1	1	1
Reserved	0	0	0
Automatic Clock Configuration	0	0	0
Reserved	0	0	0
LKD_WINCNT_MAX	5	5	5
Enable Internal LD	1	1	1
Reserved	0	0	0
LD Window Type	1	1	1
LD Digital Window Duration	2	2	2
LD Digital Timer Frequency	0	0	0
Reserved	0	0	0
Automatic Relock	0	0	0
Reserved	15	15	15
GPO_PAD_EN	0	0	0
Reserved	3	3	3
VCO Buffer and Pre-scalar	0	0	0
Reserved	55	55	55
High Frequency Ref	0	0	0
Reserved	512	512	512
CP DN gain	125	125	125
CP UP gain	125	125	125
Offset Magnitude	127	127	127

Offset up enable	1	1	1
Offset DN enable	0	0	0
VTUNE Resolution	6	6	6
Reserved	8	8	8
Force Curve	0	0	0
Auto-Calibration Disable	0	0	0
No VSPI Trigger	0	0	0
FSM/VSPI Clock Trigger	1	1	1
Reserved	0	0	0
PD/DEL_SEL	1	1	1
Reserved	0	0	0
PD_UP_EN	1	1	1
PD_DN_EN	1	1	1
CSP Mode	0	0	0
Force CP Up	0	0	0
Force CP DN	0	0	0
Reserved	496	496	496
Number of channels	0	0	0
GPO_SELECT	1	1	1
GPO Test Data	0	0	0
Prevent Auto-mux SDO	0	0	0
LDO driver always on	1	1	1
Disable PFET	0	0	0
Disable NFET	0	0	0
VCO Switch Setting	234	235	232
Auto-Calibration Busy	0	0	0
SAR Error Counts	6	8	19
SAR Error Sign	1	1	0
GPO	0	0	0
LD	0	0	0
Reserved	4697	4697	4697

Programming Code for Arduino:

```
#include <SPI.h>

const int chipSelectPin = 22;

long Outputfrequency=2000e6;
long ReferenceFrequency=20e6;

//PLL_REG0x01
long RST_CHIPEN_PIN_SELECT=0; /*1-bit (1 = take PLL enable via CEN pin,0 = take PLL enable via SPI (RST_CHIPEN_FROM_SPI) Register 0x01[1]) */
long RST_CHIPEN_FROM_SPI=1; /*1-bit (PLL enable bit of the SPI) */


//PLL_REG0x02
long RDIV=1; /*14-bits (1d-16383d. Using the reference divider requires the Analog EN Register 0x08[3] = 1) */


//PLL_REG0x03
long INT_REG=Outputfrequency/ReferenceFrequency; /*19-bits (FRAC 1d-524284d INT 16d-524287d) */

//PLL_REG0x04
long FRAC=((Outputfrequency%ReferenceFrequency)*(16777216))/ReferenceFrequency; /*24-bits (0d-(2^24-1)d) */


//PLL_REG0x05
long VCO_ID=0; /*3-bits */
long VCO_REGADDR=0; /*4-bits */
long VCO_DATA=0; /*9-bits */


//PLL_REG0x06
long Seed=2; /*2-bits (Selects the seed in fractional mode 0: 0 seed.1: LSB seed.2: 0xB29D08 seed.3: 0x50F1CD seed.) */
long FRAC_BYPASS=0; /*1-bit (0: use modulator, required for fractional mode 1: bypass modulator, required for integer mode.) */
long Initialization=7; /*3-bits (Program to 7d.) */
long SD_ENABLE=1; /*1-bit (0: disables fractional core, use for integer mode or integer mode with CSP.1: enables fractional core, required for fractional mode, or integer isolationtesting.) */
long Automatic_Clock_Conf=0; /*1-bit (Program to 0) */


//PLL_REG0x07
long LKD_WINCNT_MAX=5; /*3-bits (number of consecutive counts of divided VCO that must land inside the lock detect window to declare lock 0: 5.1: 32.2: 96.3: 256.4: 512.5: 2048.6: 8192.7: 65,535) */
long Enable_Internal_LD=1; /*1-bit () */
long LD_Window_Type=1; /*1-bit (1: digital programmable timer 0: analog one shot, nominal 10 ns window) */
long LD_Digital_Window_Dur=2; /*3-bits (0: half cycle.1: one cycle.2: two cycles.3: four cycles.4: eight cycles.5: 16 cycles.6: 32 cycles.7: 64 cycles) */
long LD_Digital_Timer_Freq=0; /*2-bits (00: fastest 11: slowest) */
long Automatic_Relock=0; /*1-bit (1: attempts to relock if lock detect fails for any reason; tries one time only) */


//PLL_REG0x08
long GPO_PAD_EN=0; /*1-bit (0: disables the LD/SDO pin 1: enables GPO port or allows a shared SPI) */
long VCO_Buffer_and_Presc=0; /*1-bit (VCO buffer and prescaler bias enable) */
long High_Frequency_Ref=0; /*1-bit (Program to 1 for XTAL > 200 MHz) */


//PLL_REG0x09
long CP_DN_gain=125; /*7-bits (0d = 0 u A. 1d = 20 u A. 2d = 40 u A... 127d = 2.54 mA) */
long CP_UP_gain=125; /*7-bits (0d = 0 u A. 1d = 20 u A. 2d = 40 u A... 127d = 2.54 mA) */
long Offset_Magnitude=127; /*7-bits (0d = 0 u A. 1d = 5 u A. 2d = 10 u A...127d = 635 u A)*/
long Offset_UP_enable=1; /*1-bit (Recommended setting = 1 in fractional mode, 0 otherwise.) */
long Offset_DN_enable=0; /*1-bit (Recommended setting = 0) */


//PLL_REG0x0A
long VTUNE_Resolution=6; /*3-bits (R divider cycles 0: 1 cycle 1: 2 cycles 2: 4 cycles ... 7: 256 cycles) */
long Force_Curve=0; /*1-bit (Program 0) */
long AutoCalibration_Disabl=0; /*1-bit (Program 0 for normal operation using VCO autocalibration) */
long No_VSPI_Trigger=0; /*1-bit (0: normal operation 1: this bit disables the serial transfers to the VCO subsystem (viaRegister 0x05)) */
long FSM_VSPI_Clock_Trigger=1; /*2-bits (These bits set the autocalibration FSM and VSPI clock (50 MHzmaximum) 0: input crystal reference 1: input crystal reference divide by 4 2: input crystal reference divide by 16 3: input crystal reference divide by 32) */


//PLL_REG0x0B
long PD_DEL_SEL=1; /*3-bits (Sets PD reset path delay (recommended setting is 001).) */
long PD_UP_EN=1; /*1-bit (Enables the PD up output.) */
long PD_DN_EN=1; /*1-bit (Enables the PD down output) */
```

```
long CSP_Mode=0; /*2-bits (Extra current is driven into the loop filter when the phase erroris larger than the following: 0 = disabled. 1 = 5.4 ns. 2 = 14.4 ns. 3 = 24.1 ns) */
long Force_CP_UP=0; /*1-bit (Forces CP up output to turn on; use for test only.) */
long Force_CP_DN=0; /*1-bit (Forces CP down output to turn on; use for test only.) */


//PLL_REG0x0C
long Number_of_channels=0; /*14-bits (The comparison frequency divided by the correction rate must be an integer.Frequencies at exactly the correction rate have zero frequency error. 0: disabled. 1: disabled. 2:16383d (0x3FFF).) */


//PLL_REG0x0F
long GPO_SELECT=1; /*5-bits (The signal selected here is an output to the SDO pin when the SDOpin is enable via Register 0x08[5]0: data from Register 0x0F[5]1: lock detect output2: lock detect trigger3: lock detect window output4: ring oscillator test5: pull-up hard from CSP6: pull-down hard from CSP7: reserved
//8: reference buffer output9: reference divider output10: VCO divider output11: modulator clock from VCO divider12: auxiliary clock13: auxiliary SPI clock14: auxiliary SPI enable15: auxiliary SPI data output16: PD down17: PD up18: SD3 clock delay19: SD3 core clock20: autostrobe integer write21: autostrobe fractional write
//22: autostrobe auxiliary SPI23: SPI latch enable24: VCO divider sync reset25: seed load strobe26 to 29: not used30: SPI output buffer enable31: soft reset, RST) */
long GPO_Test_Data=0; /*1-bit () */
long Prevent_Automux_SDO=0; /*1-bit (1: outputs GPO data only0: automuxes between SDO and GPO data) */
long LDO_driver_always_on=1; /*1-bit (1: LD_SDO pin driver always on0: LD_SDO pin driver only on during SPI read cycle) */
long Disable_PFET=0; /*1-bit () */
long Disable_NFET=0; /*1-bit () */


//VCO_REG0x00
long CAL=0; /*1-bit (VCO tune voltage is redirected to a temperature compensated calibration voltage) */
long CAPS = 0; /*8-bits (VCO subband selection 0: maximum frequency 1111 1111: minimum frequency) */


//VCO_REG0x01
long Master_Enable_VCO=1; /*1-bit (0: all VCO subsystem blocks are turned off.) */
long VCO_Enable=1; /*1-bit (Enables VCOs) */
long PLL_Buffer_Enable=1; /*1-bit (Enables PLL buffer to N divider) */
long I_O_master_Enable=1; /*1-bit (Enables output stage and the output divider. It does not enable/disable the VCO.) */
long Output_Stage_Enable=1; /*1-bit (Output stage enable) */


//VCO_REG0x02
long RD_divide_ratio=1; /*6-bits (0: mutes the output when VCO_REG 0x03[8:7] = 0d.1: fo.2: fo/2.3: invalid, defaults to 2.4: fo/4.5: invalid, defaults to 4.6: fo/6.60: fo/60.61: invalid, defaults to 60.62: fo/62 > 62 invalid, defaults to 62) */


//VCO_REG0x03
long Programmable_Performan=3; /*2-bits (Selects output noise floor performance level at a cost of increasedcurrent consumption.01: low current consumption mode.11: high performance mode.Other states (00 and 10) not supported.) */
long RF_N_Enable=1; /*1-bit (Enables the output on RF_N pin.) */
long RF_P_Enable=1; /*1-bit (Enables the output on RF_N pin.) */
long Return_loss=0; /*1-bit (0: return loss = -5 dB typical (highest output power).1: return loss = -10 dB typical.) */
long Mute_mode=0; /*2-bits (00: enables mute when the divide ratio, VCO_REG 0x02[5:0] = 0. Thisenables the HMC832 to be backwards compatible to the HMC830 mutefunction.01: during VCO calibration (see the VCO Calibration section for moredetails).10: not supported.11: mute all RF outputs (unconditional)) */


//VCO_REG0x04
long Initialization2=201; /*9-bits (Reserved) */


//VCO_REG0x07
long Output_gain=11; /*4-bits (Output stage gain control in 1 dB steps0d: 0 dB gain1d: 1 dB gain2d: 2 dB gain10d: 10 dB gain11d: 11 dB gain) */
long Initialization3=1; /*1-bit (Program to 1d) */

double difference(double a,double b)
{if(a>b){return (a-b);}
   else {return (b-a);}
}


long REG_00=0x20;
long REG_01=(RST_CHIPEN_FROM_SPI<<1)|(RST_CHIPEN_PIN_SELECT);
long REG_02=RDIV;
long REG_03=INT_REG;
long REG_04=FRAC;
//long REG_05=;
long REG_06=(Automatic_Clock_Conf<<21)|(SD_ENABLE<<11)|(Initialization<<8)|(FRAC_BYPASS<<7)|(18<<2)|(Seed);
long
REG_07=(Automatic_Relock<<13)|(LD_Digital_Timer_Freq<<10)|(LD_Digital_Window_Dur<<7)|(LD_Window_Type<<6)|(Enable_Internal_LD<<3)|(LKD_WINCNT_MAX);
long REG_08=(3<<22)|(High_Frequency_Ref<<21)|(55<<11)|(VCO_Buffer_and_Presc<<10)|(11<<6)|(GPO_PAD_EN<<5)|(31);
long REG_09=(Offset_DN_enable<<22)|(Offset_UP_enable<<21)|(Offset_Magnitude<<14)|(CP_UP_gain<<7)|(CP_DN_gain);
```

```
long REG_0a=(FSM_VSPI_Clock_Trigger<<13)|(No_VSPI_Trigger<<12)|(AutoCalibration_Disabl<<11)|(Force_Curve<<10)|(64<<3)|(VTUNE_Resolution);
long REG_0b=(496<<11)|(Force_CP_DN<<10)|(Force_CP_UP<<9)|(CSP_Mode<<7)|(PD_DN_EN<<6)|(PD_UP_EN<<5)|(PD_DEL_SEL);
long REG_0c=Number_of_channels;
//long REG_0d=;
//long REG_0e=;
long REG_0f=(Disable_NFET)|(Disable_PFET)|(LDO_driver_always_on)|(Prevent_Automux_SDO)|(GPO_Test_Data)|(GPO_SELECT);
//long REG_10=;
//long REG_11=;
//long REG_12=;
//long REG_13=;
//long REG_14=;
long REG_50=(CAPS<<1)|(CAL);
long REG_51=(1<<8)|(3<<6)|(Output_Stage_Enable<<5)|(1<<4)|(I_O_master_Enable<<3)|(PLL_Buffer_Enable<<2)|(VCO_Enable<<1)|(Master_Enable_VCO);
long REG_52=RD_divide_ratio;
long REG_53=(Mute_mode<<7)|(Return_loss<<5)|(1<<4)|(RF_P_Enable<<3)|(RF_N_Enable<<2)|(Programmable_Performan);
long REG_54=201;
long REG_55=170;
long REG_56=255;
long REG_57=(4<<5)|(Initialization3<<4)|(Output_gain);


SPISettings HMCModeSettings(100000, MSBFIRST, SPI_MODE0);
SPISettings OpenModeSettings(100000, MSBFIRST, SPI_MODE3);

void setup() {
  delay(2000);
  Serial.begin(9600);
  pinMode(chipSelectPin, OUTPUT);
  pinMode(chipSelectPin, HIGH);
  SPI.begin();
  delay(3000);
  Serial.println("Initialized");


/*
double ans=0;
for(double f=0;f<=16777216;f++){
for(double r=1;r<=16383;r++){
for(double i=1;i<=524284;i++){
for(double d=1;d<=62;d++){
ans=((double(ReferenceFrequency)/r)*(i+(f/16777216)));
if((ans>=1.5e9)&&(ans<=3e9)&&( difference((ans/d),double(Outputfrequency))<1e4)  )
{
RDIV = long(r);
INT_REG = long(i);
FRAC = long(f);if(FRAC>0){FRAC_BYPASS=0;}
RD_divide_ratio = long(d);
CAPS = long(((3e9 - ((double(ReferenceFrequency)/r) * (double(INT_REG) + (double(FRAC) / 16777216)))) * 255) / (1.5e9));
r=16384;
i=524285;
f=16777217;
d=63;
}
}
}
}
}


REG_02=RDIV;
REG_03=INT_REG;
REG_04=FRAC;
REG_06=(Automatic_Clock_Conf<<21)|(SD_ENABLE<<11)|(Initialization<<8)|(FRAC_BYPASS<<7)|(18<<2)|(Seed);
REG_50=(CAPS<<1)|(CAL);
REG_52=RD_divide_ratio;
*/

}

void loop() {


/*Comment out next line for GPS (1621.84 MHz)*/
// INT_REG=81; REG_03=INT_REG; FRAC=1543504; REG_04=FRAC;
/* Comment out next line for Beidou (1607.196 MHz)*/
// INT_REG=80; REG_03=INT_REG; FRAC=6036442; REG_04=FRAC;
/* Comment out next line for GLONASS (1649 MHz)*/
//INT_REG=82; REG_03=INT_REG; FRAC=7549747; REG_04=FRAC;
```

```
REG_50=(REG_50<<7)|(0<<3)|0;
REG_51=(REG_51<<7)|(1<<3)|0;
REG_52=(REG_52<<7)|(2<<3)|0;
REG_53=(REG_53<<7)|(3<<3)|0;
REG_54=(REG_54<<7)|(4<<3)|0;
REG_55=(REG_55<<7)|(5<<3)|0;
REG_56=(REG_56<<7)|(6<<3)|0;
REG_57=(REG_57<<7)|(7<<3)|0;

  for(int i=0;i<3;i++)
  {
    HMCModeWriteRegister(0x0, REG_00);
    HMCModeWriteRegister(0x1, REG_01);
    HMCModeWriteRegister(0x2, REG_02);
    HMCModeWriteRegister(0x5, REG_51);
    HMCModeWriteRegister(0x5, REG_52);
    HMCModeWriteRegister(0x5, REG_53);
    HMCModeWriteRegister(0x5, REG_54);
    HMCModeWriteRegister(0x5, REG_55);
    HMCModeWriteRegister(0x5, REG_56);
    HMCModeWriteRegister(0x5, REG_57);
    HMCModeWriteRegister(0x5, REG_50);
    HMCModeWriteRegister(0x6, REG_06);
    HMCModeWriteRegister(0x7, REG_07);
    HMCModeWriteRegister(0x8, REG_08);
    HMCModeWriteRegister(0x9, REG_09);
    HMCModeWriteRegister(0xa, REG_0a);
    HMCModeWriteRegister(0xb, REG_0b);
    HMCModeWriteRegister(0xc, REG_0c);
    HMCModeWriteRegister(0xf, REG_0f);
    HMCModeWriteRegister(0x3, REG_03);
    HMCModeWriteRegister(0x4, REG_04);
    Serial.println("HMCModeWrite Complete");
    delay(10000);
  }
  while(1){}
}

void OpenModeWriteRegister(long thisRegister, long thisValue) {

  thisRegister = thisRegister<<3;
  thisValue = thisValue<<8;
  long data = thisRegister|thisValue;
  uint8_t B31_B24 = (data & 0xFF000000)>>24;
  uint8_t B23_B16 = (data & 0xFF0000)>>16;
  uint8_t B15_B8  = (data & 0xFF00)>>8;
  uint8_t B7_B0   = (data & 0xFF);

  // take the chip select low to select the device:
  SPI.beginTransaction(OpenModeSettings);
  digitalWrite(chipSelectPin, LOW);

  SPI.transfer(B31_B24);
  SPI.transfer(B23_B16);
  SPI.transfer(B15_B8);
  SPI.transfer(B7_B0);

  SPI.endTransaction();
  digitalWrite(chipSelectPin, HIGH);

  delay(10);
  }

void HMCModeWriteRegister(long thisRegister, long thisValue) {

  thisRegister = thisRegister<<24;
  thisValue = thisValue;
  long data = thisRegister|thisValue;
  uint8_t B31_B24 = (data & 0b11111100000000000000000000000000)>>23;
  uint8_t B23_B16 = (data & 0b00000011111111100000000000000000)>>15;
  uint8_t B15_B8  = (data & 0b00000000000000011111111110000000)>>7;
  uint8_t B7_B0   = (data & 0b00000000000000000000000001111111)<<1;
```

```
// take the chip select low to select the device:
SPI.beginTransaction(HMCModeSettings);
digitalWrite(chipSelectPin, HIGH);
SPI.transfer(B31_B24);
SPI.transfer(B23_B16);
SPI.transfer(B15_B8);
SPI.transfer(B7_B0);
SPI.endTransaction();
digitalWrite(chipSelectPin, LOW);

delay(10);
}
```

C++ Code for Conversion from 24-bit Register values to PLL/ VCO variables:

```
#include <iostream>
using namespace std;

int main(){
/*Following PLL Registers must be read using HMC832 Evaluation Kit Software after Evaluation Kit is locked at desired frequency*/
        long R_00=0x20;
        long R_01=0x2;
        long R_02=0x1;
        long R_03=0x2A;
        long R_04=0x666666;
        long R_05=0x0;
        long R_06=0xF4A;
        long R_07=0x14D;
        long R_08=0xC1BEFF;
        long R_09=0x3FFEFD;
        long R_0A=0x2046;
        long R_0B=0xF8061;
        long R_0C=0x0;
        long R_0F=0x81;
        long R_10=0x20;
        long R_11=0x7FFFF;
        long R_12=0x0;
        long R_13=0x1259;



        cout<<"RST_CHIPEN_PIN_SELECT "<<((R_01&(0b00000000000000000000000001))>>0)<<endl;
        cout<<"RST_CHIPEN_FROM_SPI    "<<((R_01&(0b00000000000000000000000010))>>1)<<endl;
        cout<<"Reserved               "<<((R_01&(0b00000000000000011111111100))>>2)<<endl<<endl;
        cout<<"RDIV                   "<<((R_02&(0b00000000001111111111111111))>>0)<<endl<<endl;
        cout<<"INT_REG                "<<((R_03&(0b00000111111111111111111111))>>0)<<endl<<endl;
        cout<<"FRAC                   "<<((R_04&(0b11111111111111111111111111))>>0)<<endl<<endl;
        cout<<"VCO_ID                 "<<((R_05&(0b00000000000000000000000111))>>0)<<endl;
        cout<<"VCO_REGADDR            "<<((R_05&(0b00000000000000000001111000))>>3)<<endl;
        cout<<"VCO_DATA               "<<((R_05&(0b11111111111111111110000000))>>7)<<endl<<endl;
        cout<<"Seed                   "<<((R_06&(0b00000000000000000000000011))>>0)<<endl;
        cout<<"Reserved               "<<((R_06&(0b00000000000000000001111100))>>2)<<endl;
        cout<<"FRAC_BYPASS            "<<((R_06&(0b00000000000000000010000000))>>7)<<endl;
        cout<<"Initialization         "<<((R_06&(0b00000000000000011100000000))>>8)<<endl;
        cout<<"SD_ENABLE              "<<((R_06&(0b00000000000000100000000000))>>11)<<endl;
        cout<<"Reserved               "<<((R_06&(0b00011111111111000000000000))>>20)<<endl;
        cout<<"Automatic Clock Conf   "<<((R_06&(0b00100000000000000000000000))>>21)<<endl;
        cout<<"Reserved               "<<((R_06&(0b01000000000000000000000000))>>22)<<endl<<endl;
        cout<<"LKD_WINCNT_MAX         "<<((R_07&(0b00000000000000000000000111))>>0)<<endl;
        cout<<"Enable Internal LD     "<<((R_07&(0b00000000000000000000001000))>>3)<<endl;
        cout<<"Reserved               "<<((R_07&(0b00000000000000000000110000))>>4)<<endl;
        cout<<"LD Window Type         "<<((R_07&(0b00000000000000000001000000))>>6)<<endl;
        cout<<"LD Digital Window Dur  "<<((R_07&(0b00000000000000001110000000))>>7)<<endl;
        cout<<"LD Digital Timer Freq  "<<((R_07&(0b00000000000001100000000000))>>10)<<endl;
        cout<<"Reserved               "<<((R_07&(0b00000000000010000000000000))>>12)<<endl;
        cout<<"Automatic Relock       "<<((R_07&(0b00000000000100000000000000))>>13)<<endl<<endl;
        cout<<"Reserved               "<<((R_08&(0b00000000000000000000001111))>>0)<<endl;
        cout<<"GPO_PAD_EN             "<<((R_08&(0b00000000000000000010000))>>5)<<endl;
        cout<<"Reserved               "<<((R_08&(0b00000000000000000111100000))>>6)<<endl;
        cout<<"VCO Buffer and Presc   "<<((R_08&(0b00000000000000001000000000))>>10)<<endl;
        cout<<"Reserved               "<<((R_08&(0b00011111111111000000000000))>>11)<<endl;
        cout<<"High Frequency Ref     "<<((R_08&(0b00100000000000000000000000))>>12)<<endl;
        cout<<"Reserved               "<<((R_08&(0b11000000000000000000000000))>>13)<<endl<<endl;
        cout<<"CP DN gain             "<<((R_09&(0b00000000000000000111111111))>>0)<<endl;
        cout<<"CP UP gain             "<<((R_09&(0b00000000011111111000000000))>>7)<<endl;
        cout<<"Offset Magnitude       "<<((R_09&(0b00111111100000000000000000))>>14)<<endl;
        cout<<"Offset up enable       "<<((R_09&(0b01000000000000000000000000))>>21)<<endl;
        cout<<"Offset DN enable       "<<((R_09&(0b10000000000000000000000000))>>22)<<endl<<endl;
        cout<<"VTUNE Resolution       "<<((R_0A&(0b00000000000000000000000111))>>0)<<endl;
        cout<<"Reserved               "<<((R_0A&(0b00000000000001111111000000))>>3)<<endl;
        cout<<"Force Curve            "<<((R_0A&(0b00000000000010000000000000))>>10)<<endl;
        cout<<"AutoCalibration Disabl "<<((R_0A&(0b00000000000100000000000000))>>11)<<endl;
        cout<<"No VSPI Trigger        "<<((R_0A&(0b00000000001000000000000000))>>12)<<endl;
        cout<<"FSM/VSPI Clock Trigger "<<((R_0A&(0b00000000110000000000000000))>>13)<<endl;
        cout<<"Reserved               "<<((R_0A&(0b00000011000000000000000000))>>15)<<endl<<endl;
        cout<<"PD/DEL_SEL             "<<((R_0B&(0b00000000000000000000000111))>>0)<<endl;
        cout<<"Reserved               "<<((R_0B&(0b00000000000000000000011000))>>3)<<endl;
        cout<<"PD_UP_EN               "<<((R_0B&(0b00000000000000000000100000))>>5)<<endl;
        cout<<"PD_DN_EN               "<<((R_0B&(0b00000000000000000001000000))>>6)<<endl;
        cout<<"CSP Mode               "<<((R_0B&(0b00000000000000000110000000))>>7)<<endl;
```

```
        cout<<"Force CP Up                "<<((R_0B&(0b00000000000001000000000))>>9)<<endl;
        cout<<"Force Cp DN                "<<((R_0B&(0b00000000000010000000000))>>10)<<endl;
        cout<<"Reserved                  "<<((R_0B&(0b11111111111100000000111))>>11)<<endl<<endl;
        cout<<"Number of channels         "<<((R_0C&(0b00000000011111111111111))>>0)<<endl<<endl;
        cout<<"GPO_SELECT                 "<<((R_0F&(0b00000000000000000011111))>>0)<<endl;
        cout<<"GPO Test Data              "<<((R_0F&(0b00000000000000000100000))>>5)<<endl;
        cout<<"Prevent Automux SDO        "<<((R_0F&(0b00000000000000001000000))>>6)<<endl;
        cout<<"LDO driver always on       "<<((R_0F&(0b00000000000000010000000))>>7)<<endl;
        cout<<"Disable PFET               "<<((R_0F&(0b00000000000000100000000))>>8)<<endl;
        cout<<"Disable NFET               "<<((R_0F&(0b00000000000001000000000))>>9)<<endl<<endl;
        cout<<"VCO Switch Setting          "<<((R_10&(0b00000000000000011111111))>>0)<<endl;
        cout<<"AutoCalibration Busy       "<<((R_10&(0b00000000000000100000000))>>8)<<endl<<endl;
        cout<<"SAR Error Counts           "<<((R_11&(0b00001111111111111111111))>>0)<<endl;
        cout<<"SAR Error Sign             "<<((R_11&(0b00010000000000000000000))>>19)<<endl<<endl;
        cout<<"GPO                        "<<((R_12&(0b00000000000000000000001))>>0)<<endl;
        cout<<"LD                         "<<((R_12&(0b00000000000000000000010))>>1)<<endl<<endl;
        cout<<"Reserved                  "<<((R_13&(0b00000011111111111111111))>>0)<<endl<<endl;


}
```

References

[1] “HMC832 Datasheet.” Analog Devices, Inc. http://www.analog.com/media/en/technical-documentation/data-sheets/HMC832.pdf

[2] “Evaluaion Kit Schematic.” Analog Devices, Inc. http://www.analog.com/media/en/technical-documentation/evaluation-documentation/hmc832lp6ge_eval_pcb_schematic.pdf

[3] “Evaluation Kit User Manual.” Analog Devices, Inc. http://www.analog.com/media/en/technical-documentation/evaluation-documentation/Hittite PLL Evaluation Kit User Manual.pdf

[4] “ADI HMC PLL VCO Evaluation Software Download.” *LTC3786: High Efficiency Li-Ion Battery-to-USB Boost Converter | Analog Devices*, www.analog.com/en/design-center/evaluation-hardware-and-software/pll-vco-evaluation-software.html.

[5] “HMC832.” *LTC3786: High Efficiency Li-Ion Battery-to-USB Boost Converter | Analog Devices*, www.analog.com/en/products/clock-and-timing/phase-locked-loop/fractional-n-pll/hmc832.html#product-evaluationkit.

[6] “SPI Mode Selection.” Analog Devices, Inc. http://www.analog.com/media/en/technical-documentation/application-notes/SPI-Mode-Selection-Application-Note.pdf

[7] “Power Up Brown Out” Analog Devices, Inc. http://www.analog.com/media/en/technical-documentation/application-notes/rf-pll-vco-power-up-brown-out-design-application-note.pdf

[8] “PLL VCO FAQ.” Analog Devices, Inc. http://www.analog.com/media/en/technical-documentation/application-notes/PLL-VCO-FAQ-20140919-application-note.pdf

[9] “Fast Frequency Hopping.” Analog Devices, Inc. http://www.analog.com/media/en/technical-documentation/application-notes/Fast-Frequency-Hopping-Application-Note.pdf

[10] “Products Viewed.” *ADIsimPLL Request for Software Form | Analog Devices*, form.analog.com/Form_Pages/RFComms/ADISimPll.aspx.