PLL Summer Research Project Report

Grant ID: SRP056-2017-EED
Proposal Title: PHASE LOCKED LOOP FREQUENCY SYNTHESIZER USING PE3336 IC
Internal Order: SRP-056
Purchasing Group: A49
PR Raiser: Affan/Iqbal (Department Coordinator)
PR Approver: Muhammad Faisal (OSPR)
Team Lead: Muhammad Shamaas
Faculty Mentor: Wasif Tanveer

Muhammad Abdullah 18100192

Hira Tariq 18100215

Muhammad Abdullah Khan 18100146

Muhammad Shamaas 18100217

Basis of research:

The aim of our research project is to make a low phase noise phase locked loop frequency synthesizer using MAX2871 IC. The frequency synthesizer must be able to produce high frequency stable output for use in RF (3kHz to 300GHz) applications.

Phase Locked Loops are used in telecommunications, computers, radio and mobile phones, satellite receivers, Global Positioning Systems, Local Multipoint Distribution Service, Multichannel Multipoint Distribution Service, Wireless Local Loop base stations, demanding terrestrial systems and other electronic applications. They are frequently used in wireless communication, primarily in frequency modulation or phase modulation transmissions, recovery of small signals that otherwise would be lost in noise, Dual Tone Multi Frequency decoders, modems, tone decoders, Atomic force microscopy and DC motor drives. Phase-locked loops are more commonly used for digital data transmission than for analog transmission.

They are used to generate, stabilize, modulate, demodulate, filter or recover a signal from a noisy communications channel where data has been interrupted. Due to their immense importance in these broad areas, there is a pressing need to improve the efficiency, phase noise, bandwidth, speed, transient response, steady state errors, output spectrum purity, power consumption, output amplitude, hold in range, pull in range, lock in range, type and order of phase locked loops.

The project focuses on making a frequency phase locked loop which overcomes some if not all of these short comings and produces stable high frequency output with desirable loop dynamics, stability, lock range, gain and phase margin, bandwidth and speed.

Fundamentals of Phase Locked Loops (PLLs)

A phase Locked Loop is used to maintain a constant phase angle relative to a reference signal. It consists of a feedback mechanism with a phase detector, loop filter and voltage controlled oscillator in forward path and a feedback divider in feedback path. The Phase Frequency Detector uses two inputs to control two current sources through flip flops. It can be used to detect frequency lock and phase lock. The output signal frequency is divided using a counter and pre-scalar and then compared with a low frequency reference signal frequency. Hence Phase Locked Loops can be used to generate stable output high frequency signals from a fixed low-frequency signal. A dual-modulus pre-scalar is a counter whose division ratio can be switched from one value to another by an external control signal. Phase noise is defined as the ratio of the noise in a 1 Hz bandwidth at a specified frequency offset to the oscillator signal amplitude at desired frequency. The loop bandwidth determines the frequency/ phase lock time while phase noise (caused by thermal noise, shot noise or flicker noise in active and passive devices) and reference spurs (caused by known clock frequencies in the signal source, power line interference, and mixer products) impact spectral purity. Integrating the phase noise plot over the desired frequency range gives the time jitter. Fractional-N feedback divider allows the resolution at the Phase Locked Loop output to be reduced to small fractions of the Phase Frequency Detector. Fractional-N Phase Locked Loops provide much lower phase noise than integer-N Phase Locked Loops but at the expense of higher spurious levels. The ADIsimPLL software allows us to construct a Phase Locked Loop by specifying the frequency requirements and selecting implementation details. The program designs a loop filter and displays phase noise, reference spurs, lock time, lock detect performance etc. for fine tuning.

<u>Technical Details:</u>

MAX2871 is a high performance low power Phase Locked Loop. By subtracting, adding, dividing or multiplying a precise and stable standard reference frequency through multiplier, mixer or divider, the synthesizer generates many combinations of the signal with the same accuracy and stability. Hence it provides a wide tuning range with no compromise on precision.

It is equipped with multiple integrated VCOs covering 3000-6000MHz. The output Frequency Divider Ratios of 1/2/4/8/16/32/64/128 are available.

It offers low phase noise and high frequency of operation, eliminating the need of additional frequency multipliers to obtain the desired output frequency. The design of active loop filter must meet the demands of transient response, loop stability, loop dynamics, damping behavior and steady state errors. It must also eliminate Frequency sidebands produced by the phase detector.

MAX2871 IC offers an integer and fractional divider and modulus pre-scalar to accomplish very precise tuning of frequency. The Voltage Controlled Oscillator frequency is divided by a pre-scalar to bring the Voltage Controlled Oscillator frequency closer to stable input frequency. The divided Voltage Controlled Oscillator frequency is compared to the reference frequency in the Phase Frequency Detectors. When the Voltage Controlled Oscillator is running slow, clock edges from the divided Voltage Controlled Oscillator frequency lag the reference clock edges. The differences in phase are detected by the phase detector, and up pulses are generated, similarly when the Voltage Controlled Oscillator is running fast, the reference clock lags the divided Voltage Controlled Oscillator frequency and down pulses are generated accordingly. This phase difference is detected and the charge pump adjusts the control voltage until the phase difference between the reference clock and the divided Voltage Controlled Oscillator clock is zero.

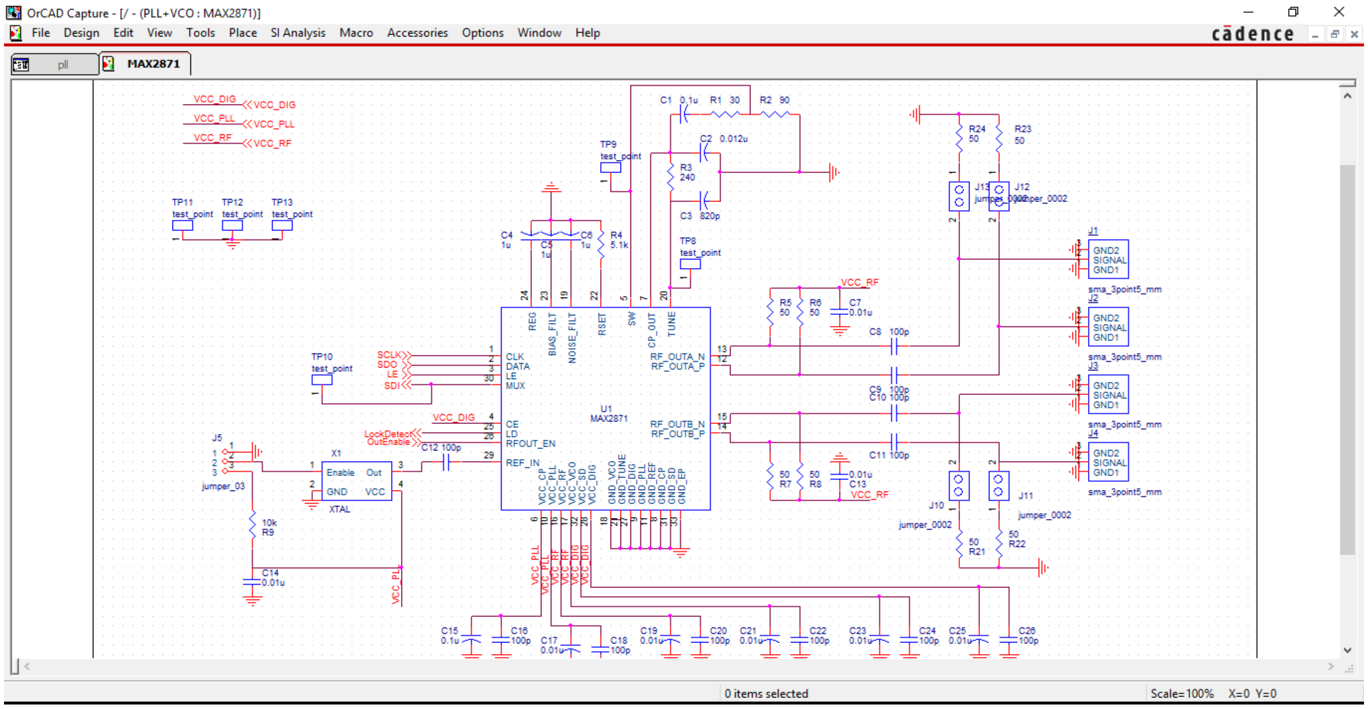
It controls the phase difference very precisely as well. The superior phase noise performance of the MAX2871 makes it ideal for applications such as Local Multipoint Distribution Service, Multichannel Multipoint Distribution Service, Wireless Local Loop base stations and demanding terrestrial systems. The phase detector is triggered by rising edges from the main Counter and the reference counter. If the divided VCO leads the divided reference in phase or frequency output pulses low. The width of pulse is directly proportional to phase offset between the two input signals. The output drives an active loop filter which controls the VCO tune voltage.

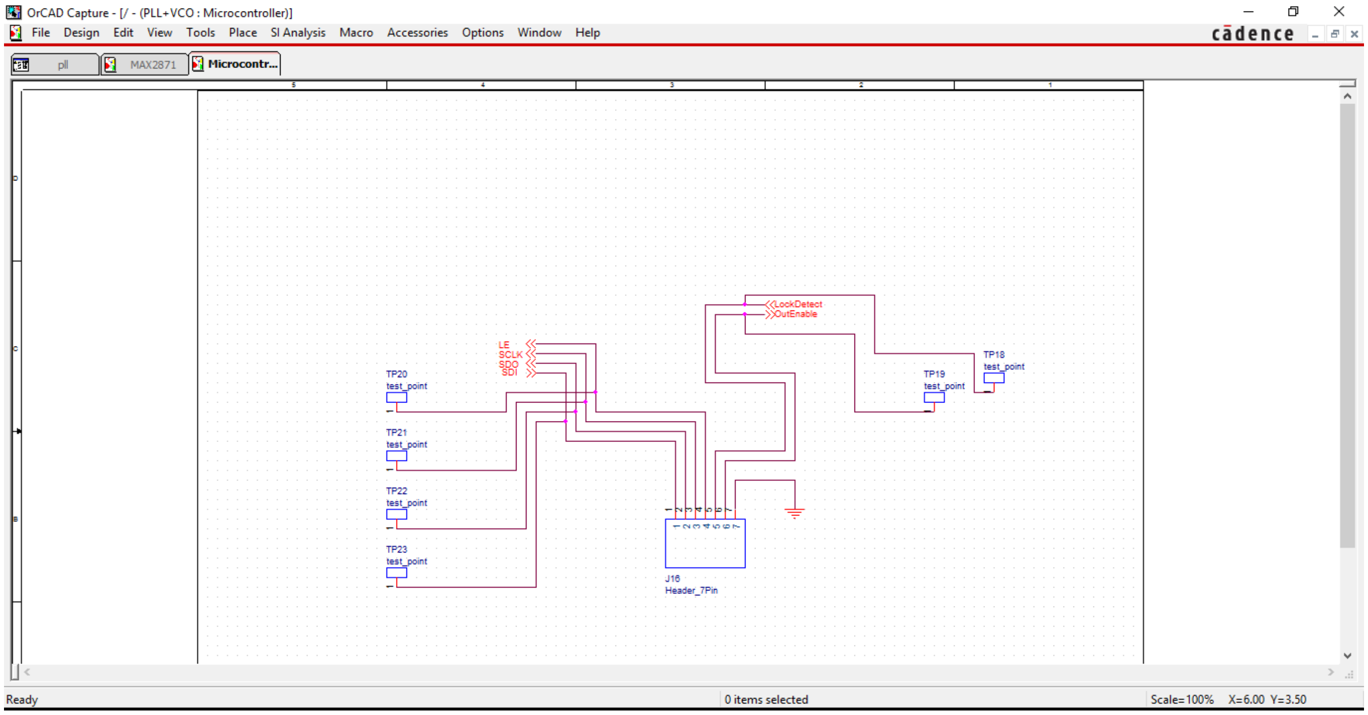
Undoubtedly MAX2871 IC offers very high bandwidth. It is serial, parallel or hardwired programmable and can be programmed for frequency synthesis from 23.5-6000 MHz. The frequency response of the IC is engineered to provide constant amplitude output at very high frequencies. The output power can be selected from -4 to +5 dBm.

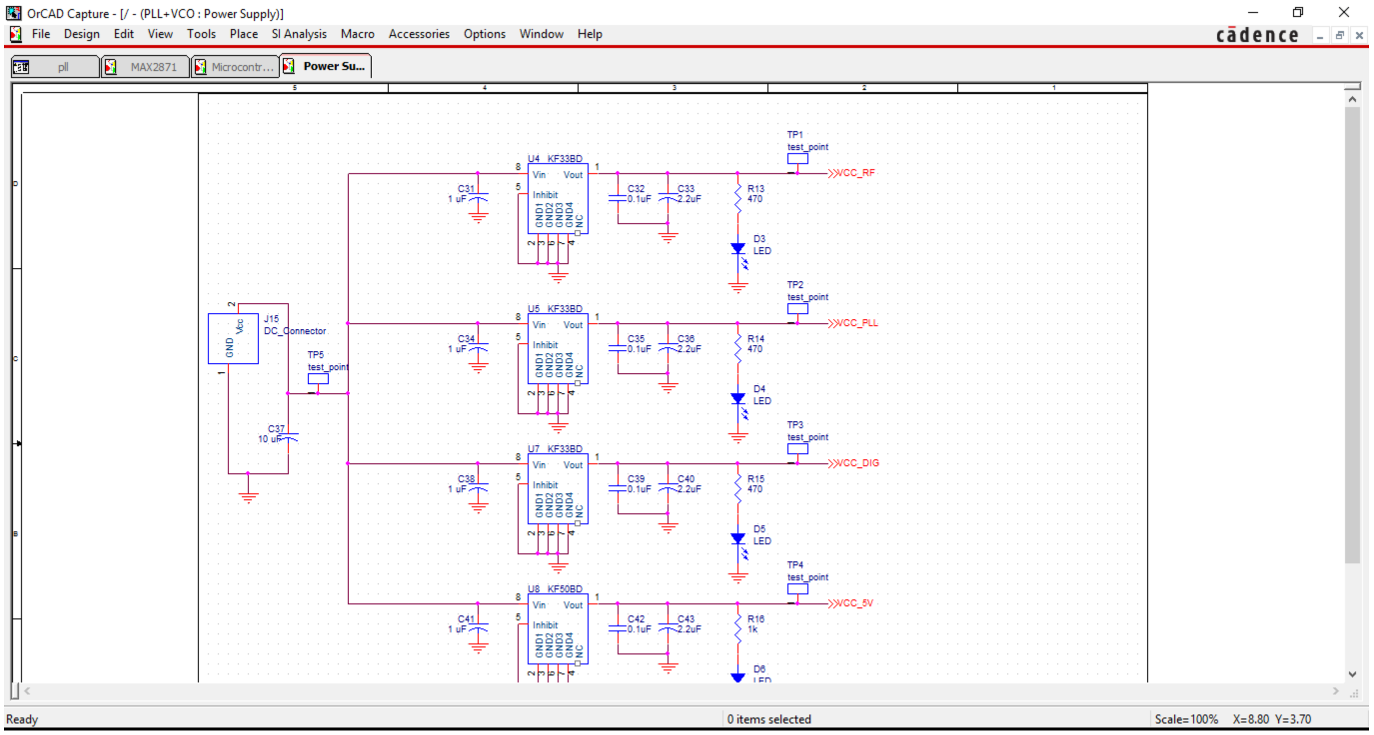
It meets the demands of speed using 20-bit shift registers, 6 and 9 bit counters, serial and parallel data buses and transistor logic gates which provide very efficient and quick transmission of control signals. The Lock Time with Fast Lock Enabled is 60us.

Often electromagnetic interference and high order harmonics corrupt the output signal of Phase Locked Loops. The output spectrum purity can be improved by designing an efficient active loop low pass filter to eliminate unwanted signal noise introduced by phase detectors and digital dividers. The In-Band Phase Noise Floor is -229 dBc/Hz.

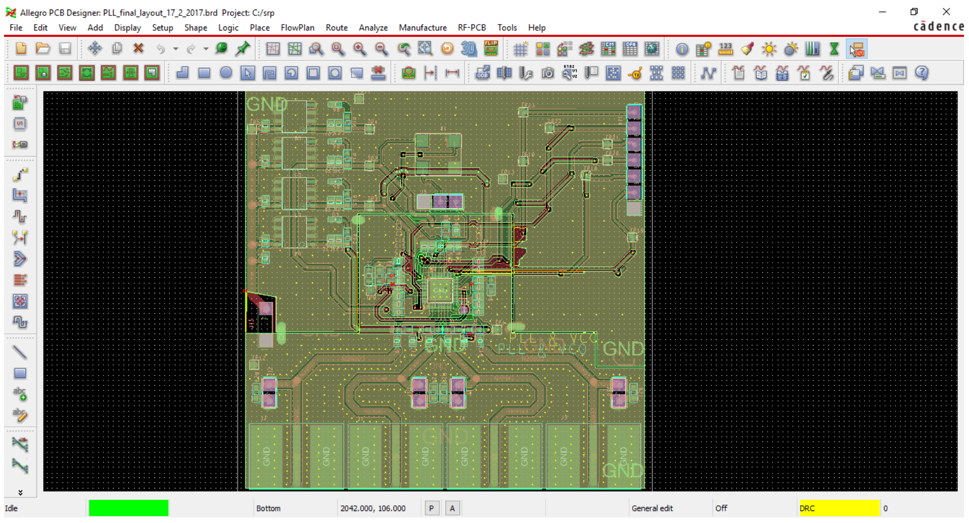
Schematic Diagrams





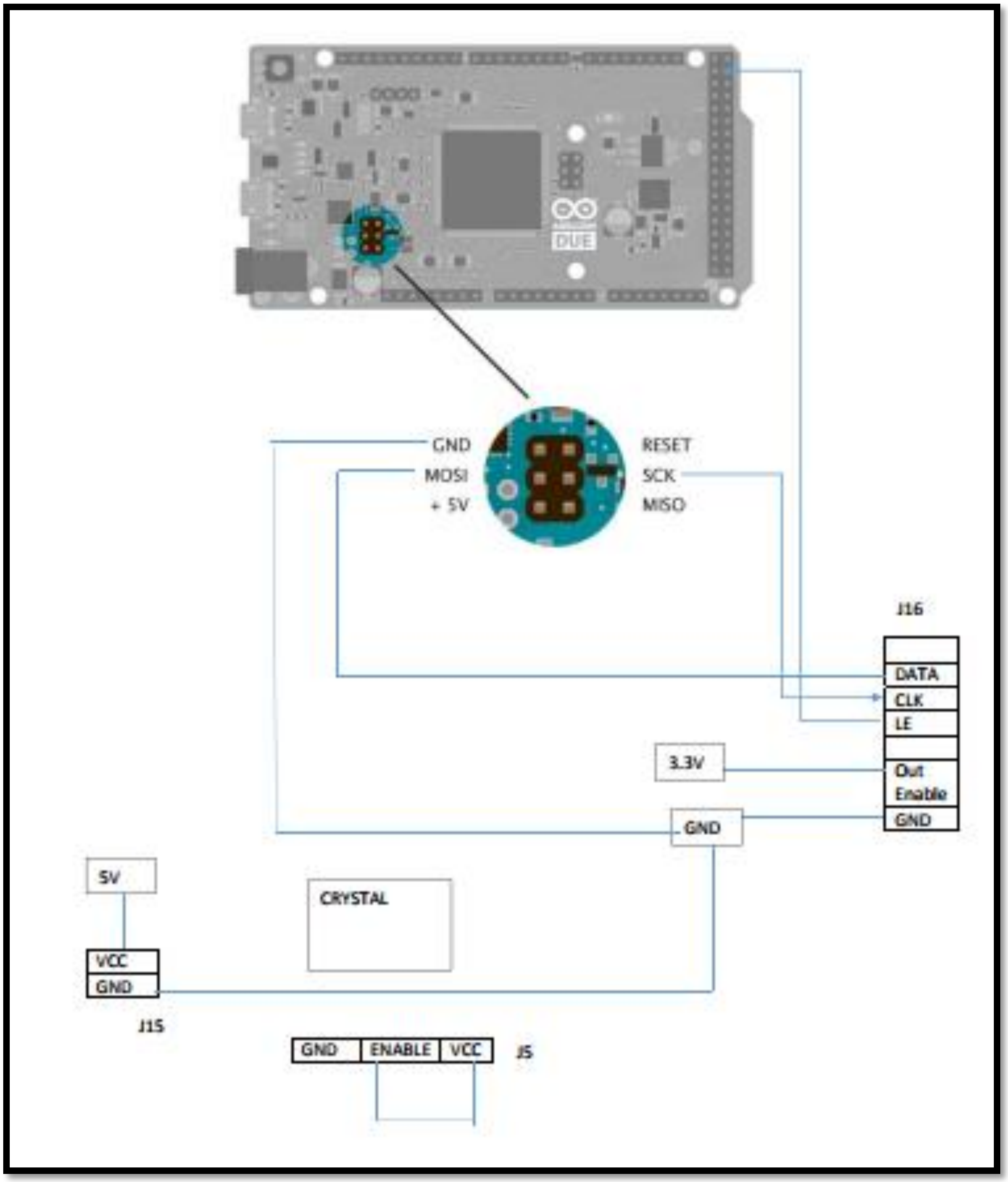


OrCAD Capture - [/ - (PLL+VCO : Power Supply)]
File Design Edit View Tools Place SI Analysis Macro Accessories Options Window Help
cādence
pll
MAX2871
Microcontr...
Power Su...
J15
DC_Connector
GND
Vcc
TP5
test_point
C37
10 uF
U4 KF33BD
U5 KF33BD
U7 KF33BD
U8 KF50BD
Vin
Vout
Inhibit
GND1 GND2 GND3 GND4 NC
C31
1 uF
C32
0.1uF
C33
2.2uF
R13
470
D3
LED
TP1
test_point
VCC_RF
C34
1 uF
C35
0.1uF
C36
2.2uF
R14
470
D4
LED
TP2
test_point
VCC_PLL
C38
1 uF
C39
0.1uF
C40
2.2uF
R15
470
D5
LED
TP3
test_point
VCC_DIG
C41
1 uF
C42
0.1uF
C43
2.2uF
R16
1k
D6
LED
TP4
test_point
VCC_5V
Ready
0 items selected
Scale=100% X=8.80 Y=3.70

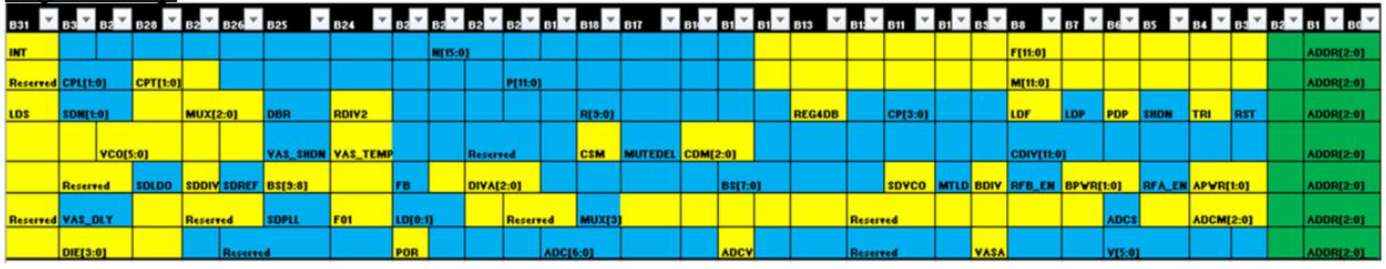


Allegro PCB Designer: PLL_final_layout_17_2_2017.brd Project: C:/srp
File Edit View Add Display Setup Shape Logic Place FlowPlan Route Analyze Manufacture RF-PCB Tools Help
cādence
GND
Idle
Bottom
2042.000, 106.000
P
A
General edit
Off
DRC
0

Pin Connections



Register Map



B31	B3	B2	B28	B2	B26	B25	B24	B2	B2	B2	B2	B1	B18	B17	B1	B1	B1	B13	B1	B11	B1	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
INT									N[15:0]														F[11:0]							ADDR[2:0]	
Reserved	CPL[1:0]		CPT[1:0]								P[11:0]												M[11:0]							ADDR[2:0]	
LDS	SDN[1:0]			MUX[2:0]		DBR	RDIV2						R[9:0]					REG4DB		CP[3:0]			LDF	LDP	PDP	SHDN	TRI	RST		ADDR[2:0]	
		VCO[5:0]				VAS_SHDN	VAS_TEMP			Reserved			CSM	MUTEDEL	CDM[2:0]								CDIV[11:0]							ADDR[2:0]	
	Reserved		SDLDO	SDDIV	SDREF	BS[9:8]		FB		DIVA[2:0]						BS[7:0]				SDVCO	MTLD	BDIV	RFB_EN	BPWR[1:0]		RFA_EN	APWR[1:0]			ADDR[2:0]	
Reserved	VAS_DLY			Reserved		SDPLL	F01	LD[0:1]			Reserved		MUX[3]						Reserved						ADCS		ADCM[2:0]			ADDR[2:0]	
	DIE[3:0]				Reserved			POR				ADC[6:0]				ADCV			Reserved			VASA			V[5:0]					ADDR[2:0]	

Sample Values of Registers from Datasheet

Reference Frequency=50MHz
Phase Detector Frequency=25MHz
2113.5MHz FRACTIONAL-N (LOW-NOISE MODE)
Register 0x00=00548050
Register 0x01=400003E9
Register 0x02=81005FC2
Register 0x03=E8000013
Register 0x04=609C80FC
Register 0x05=00400005

Variable	Value	Variable	Value	Variable	Value	Variable	Value	Variable	Value
INT	0	DBR	0	RST	0	SDREF	0	RFA_EN	1
N	169	RDIV2	1	VCO	58	BS	0	APWR	3
F	10	R	1	VAS_SHDN	0	FB	1	VAS_DLY	0
CPL	2	REG4DB	0	VAS_TEMP	0	DIVA	1	SDPLL	0
CPT	0	CP	15	CSM	0	BS2	200	F01	0
P	0	LDF	1	MUTEDEL	0	SDVCO	0	LD	1
M	125	LDP	1	CDM	0	MTLD	0	MUX_2	0
LDS	1	PDP	1	CDIV	2	BDIV	0	ADCS	0
SDN	0	SHDN	0	SDLDO	0	RFB_EN	0	ADCM	0
MUX	0	TRI	0	SDDIV	0	BPWR	3		

Arduino Code for programming the Registers

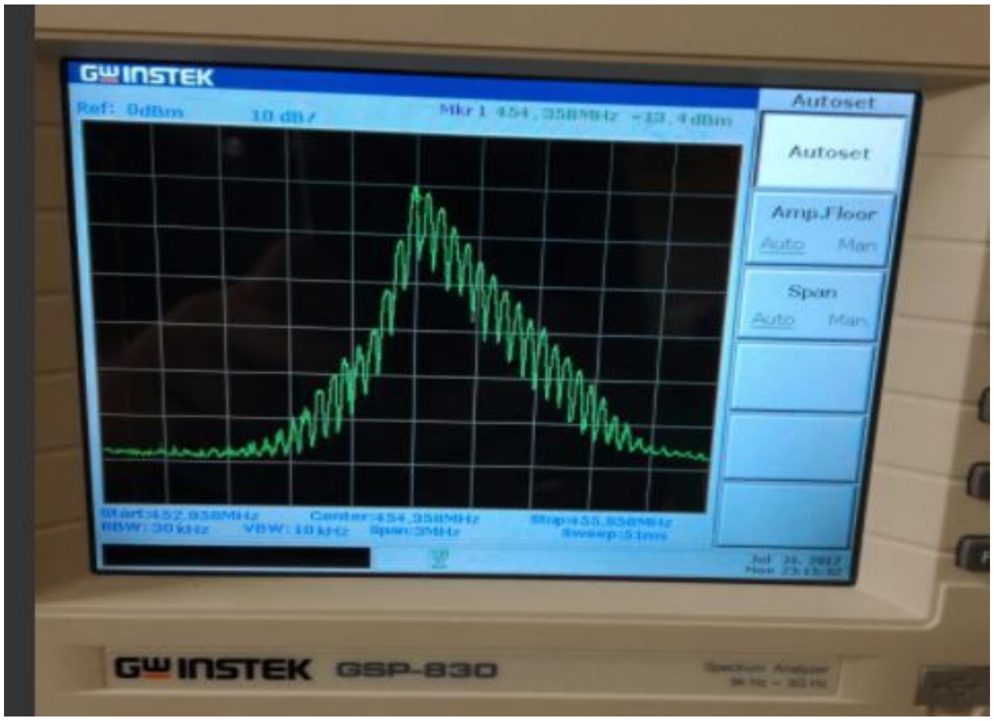
```
#include <SPI.h>
#define LEPin 22
//Data is only transferred when LE Pin (Pin 22 in this case) is low
SPISettings MAX2871_SPISettings(100000, MSBFIRST, SPI_MODE0);

void setup() {
Serial.begin(9600);
pinMode (LEPin, OUTPUT);
pinMode (LEPin, HIGH);
SPI.begin();
delay(5000);
Serial.println("Initialized");

}

void loop() {
//Program Registers in the order 0x05->0x04->0x03->0x02->0x01->0x00
//Program all registers twice upon power up


double f_desired = 2.1135e9;

int INT=0,N=120,F=1;
int CPL=2,CPT=0,P=0,M=4095;
int LDS=1,SDN=0,MUX=0,DBR=0,RDIV2=1,R=1,REG4DB=0,CP=15,LDF=1,LDP=1,PDP=1,SHDN=0,TRI=0,RST=0;
int VCO=50,VAS_SHDN=0,VAS_TEMP=0,CSM=0,MUTEDEL=0,CDM=0,CDIV=2;
intSDLDO=0,SDDIV=0,SDREF=0,BS=0,FB=1,DIVA=0,BS2=200,SDVCO=0,MTLD=0,BDIV=0,RFB_EN=0,BPWR=3,RFA_EN=1,APWR=3
int VAS_DLY=0,SDPLL=0,F01=0,LD=1,MUX_2=0,ADCS=0,ADCM=0;
int B31_B24,B23_B16,B15_B8,B7_B0;
/*
bool match=0;double ans=0,ans2=0;
   double f_ref=50e6,f_pfd,old_DBR=0,old_R=1,old_RDIV2=1,f_vco,f_RFOUTA=1,old_DIVA=0,old_N=120,old_F=1,old_M=4095,old_FB=1;
    double new_diff=0,old_diff=1e6,new_DBR=0,new_R=0,new_N=0,new_F=0,new_M=0,new_RDIV2=0,new_FB,new_DIVA;

    if (f_desired<6e9 && f_desired>=3e9){old_DIVA=1;}
    if (f_desired<3e9 && f_desired>=1.5e9){old_DIVA=2;}
    if (f_desired<1.5e9 && f_desired>=0.75e9){old_DIVA=4;}
    if (f_desired<0.75e9 && f_desired>=0.375e9){old_DIVA=8;}
    if (f_desired<0.375e9 && f_desired>=0.1875e9){old_DIVA=16;}
    if (f_desired<0.1875e9 && f_desired>=0.09375e9){old_DIVA=32;}
    if (f_desired<0.09375e9 && f_desired>=0.046875e9){old_DIVA=64;}
    if (f_desired<0.046875e9 && f_desired>=0.0234325e9){old_DIVA=128;}

    f_pfd=f_ref*((1+old_DBR)/(old_R*(1+old_RDIV2)));
    f_vco=f_desired*old_DIVA;

    for (old_N=19;old_N<=4091;old_N+=1)
          {
                for (old_F=1;old_F<=4095;old_F+=1)
                 {
                       for (old_M=2;old_M<=4095;old_M+=1)
                       {

                             ans=old_N+(old_F/old_M);
                             ans2=f_vco/f_pfd;
                             if(old_FB==1){
                             if((int(ans))==(int(ans2)))
                             {match=1;}}

                             if(old_FB==0&&old_DIVA<=16){
                             if((int(ans))==(int(ans2/old_DIVA)))
                             {match=1;}}

                             if(old_FB==0&&old_DIVA>16){
                             if((int(ans))==(int(ans2/16)))
                             {match=1;}}
```

```
                    if (match)
            {match=0;old_diff=new_diff;new_N=old_N;new_F=old_F;new_M=old_M;old_FB=1;old_M=4095;old_F=4095;old_N=4091;old_R
            DIV2=1;old_R=1023;old_DBR=1;
            }
        }
      }
   }


N=int(new_N);
F=int(new_F);
M=int(new_M);
DIVA=int(old_DIVA);

if (DIVA==1){DIVA=0;}
if (DIVA==2){DIVA=1;}
if (DIVA==4){DIVA=2;}
if (DIVA==8){DIVA=3;}
if (DIVA==16){DIVA=4;}
if (DIVA==32){DIVA=5;}
if (DIVA==64){DIVA=6;}
if (DIVA==128){DIVA=7;}
*/
B7_B0=5|(ADCM<<3)|(ADCS<<6);
B15_B8=0;
B23_B16=(MUX_2<<2)|(LD<<6);
B31_B24=F01|(SDPLL<<1)|(VAS_DLY<<5);
writeRegister(B31_B24,B23_B16,B15_B8,B7_B0);
Serial.println('5');

B7_B0=4|(APWR<<3)|(RFA_EN<<5)|(BPWR<<6);
B15_B8=RFB_EN|(BDIV<<1)|(MTLD<<2)|(SDVCO<<3)|((BS2&15)<<4);
B23_B16=((BS2&240)>>4)|(DIVA<<4)|(FB<<7);
B31_B24=BS|(SDREF<<2)|(SDDIV<<3)|(SDLDO<<4)|96;
writeRegister(B31_B24,B23_B16,B15_B8,B7_B0);
Serial.println('4');

B7_B0=3|((CDIV&31)<<3);
B15_B8=((CDIV&4064)>>5)|((CDM&1)<<7);
B23_B16=((CDM&2)>>1)|(MUTEDEL<<1)|(CSM<<2);
B31_B24=VAS_TEMP|(VAS_SHDN<<1)|(VCO<<2);
writeRegister(B31_B24,B23_B16,B15_B8,B7_B0);
Serial.println('3');

B7_B0=2|(RST<<3)|(TRI<<4)|(SHDN<<5)|(PDP<<6)|(LDP<<7);
B15_B8=LDF|(CP<<1)|(REG4DB<<5)|((R&3)<<6);
B23_B16=(R&1020)>>2;
B31_B24=RDIV2|(DBR<<1)|(MUX<<2)|(SDN<<5)|(LDS<<7);
writeRegister(B31_B24,B23_B16,B15_B8,B7_B0);
Serial.println('2');

B7_B0=(((M&31)<<3)|1);
B15_B8=((M&4064)>>5)|((P&1)<<7);
B23_B16=(P&510)>>1;
B31_B24=((P&3584)>>9)|(CPT<<3)|(CPL<<5);
writeRegister(B31_B24,B23_B16,B15_B8,B7_B0);
Serial.println('1');

B7_B0=(F&31)<<3;
B15_B8=((N&1)<<7)|(F&4064)>>5;
B23_B16=(N&510)>>1;
B31_B24=((N&65024)>>9)|(INT<<7);
writeRegister(B31_B24,B23_B16,B15_B8,B7_B0);
Serial.println('0');

while(1){
delay(10000);
N=N+1;
B7_B0=(F&31)<<3;
B15_B8=((N&1)<<7)|(F&4064)>>5;
```

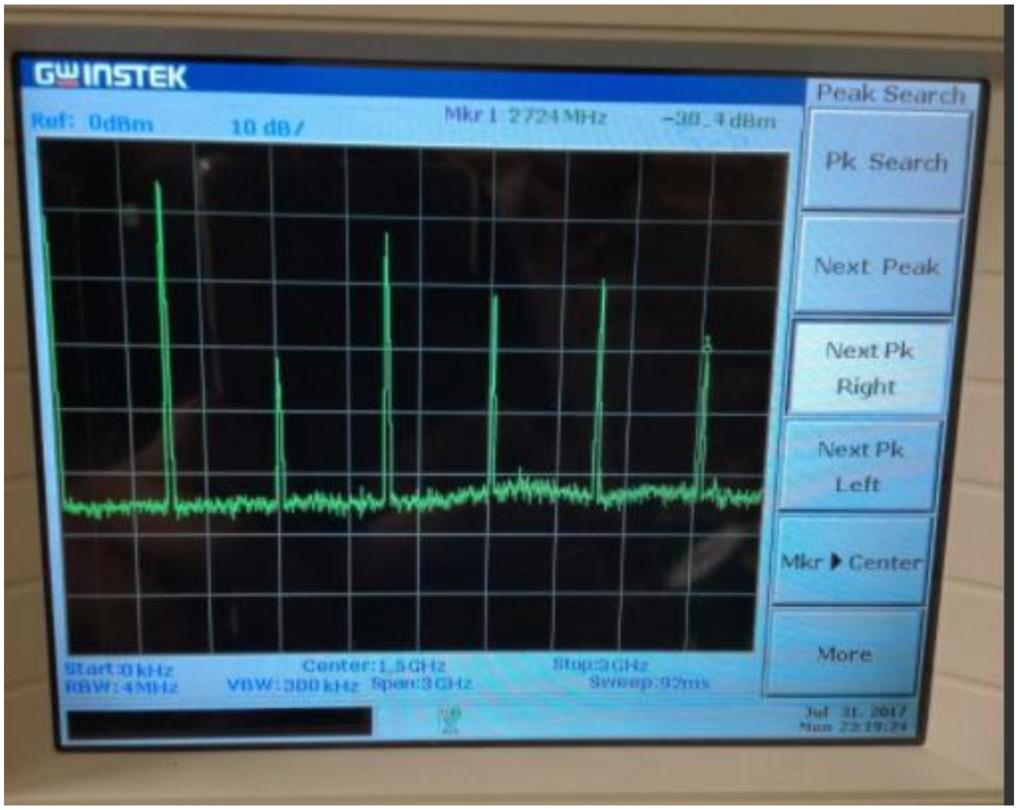
```
B23_B16=(N&510)>>1;
B31_B24=((N&65024)>>9)|(INT<<7);
writeRegister(B31_B24,B23_B16,B15_B8,B7_B0);
Serial.println('0');
}

}


void writeRegister(uint8_t b31_b24,uint8_t b23_b16,uint8_t b15_b8,uint8_t b7_b0){

digitalWrite(LEPin,LOW);
SPI.beginTransaction(MAX2871_SPISettings);
SPI.transfer(b31_b24);
SPI.transfer(b23_b16);
SPI.transfer(b15_b8);
SPI.transfer(b7_b0);
digitalWrite(LEPin,HIGH);
SPI.endTransaction();
delay(30);

}
```

Results

500MHz Output Spectrum



Harmonics Spectrum

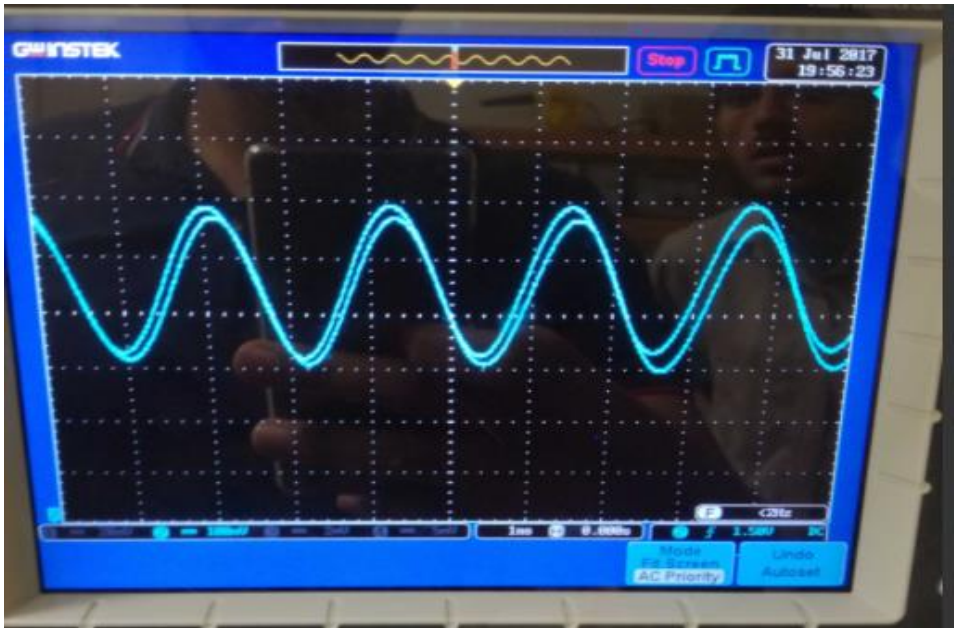


Ref: 0dBm 10 dB/ Mkr1 2724 MHz -38.4 dBm

Start:0 kHz Center:1.5 GHz Stop:3 GHz
RBW:4 MHz VBW:300 kHz Span:3 GHz Sweep:92ms

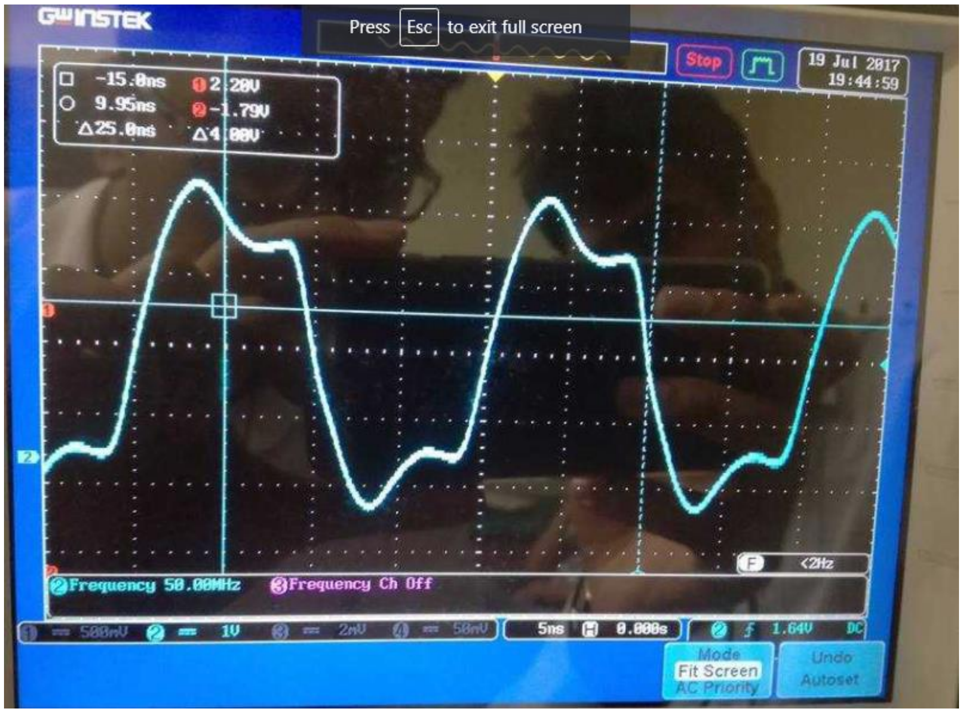
Jul 31, 2017
Mon 23:19:24

450MHz Oscilloscope Trace

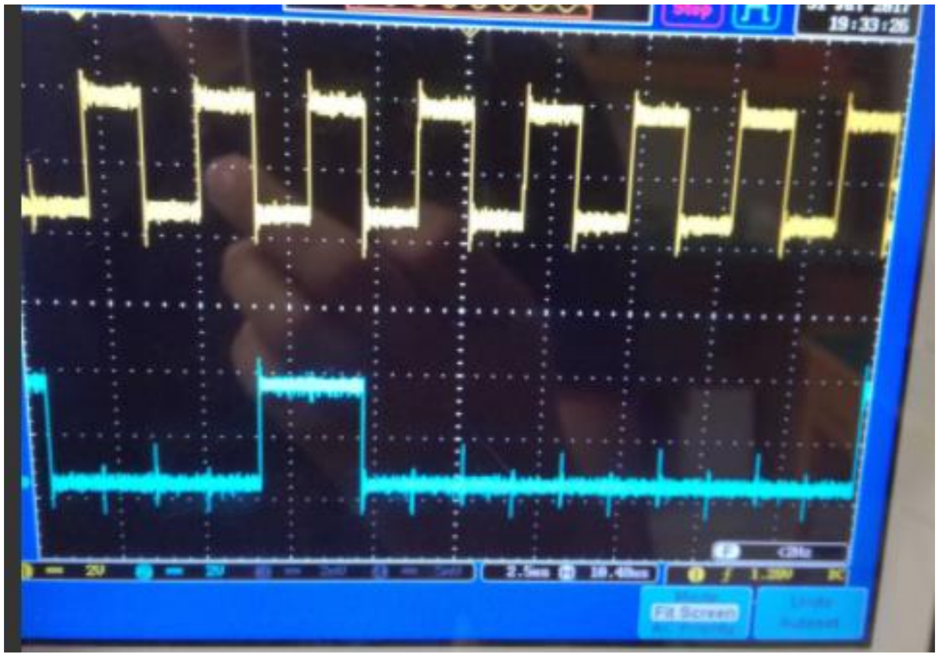


31 Jul 2017
19:56:23

50MHz Reference Crystal Oscilloscope Trace



SPI Clock/Data Pulses



Register values and corresponding output frequencies for 3021.4-3512 MHz LO Bank.

N	***F***	***M***	***Frequency(MHz)***
118	1	4095	2950.0794
119	1	4095	2974.5233
120	1	4095	3000.0794
121	1	4095	3024.5238
122	1	4095	3050.0794
123	1	4095	3075.6394
124	1	4095	3098.9683
125	1	4095	3124.5238
126	1	4095	3150.0794
127	1	4095	3174.5238
128	1	4095	3200.0794
129	1	4095	3225.6394
130	1	4095	3252.3016
131	1	4095	3275.6349
132	1	4095	3300.0794
133	1	4095	3325.6349
134	1	4095	3351.1905
135	1	4095	3375.6349
136	1	4095	3420.0794
137	1	4095	3423.4127
138	1	4095	3451.1905
139	1	4095	3482.3016
140	1	4095	3515.6349
141	1	4095	3532.3016

For each case
INT=0, CPL=2,CPT=0,P=0,
LDS=1,SDN=0,MUX=6,DBR=0,RDIV2=1,R=1,REG4DB=0,CP=15,LDF=1,LDP=1,PDP=1,SHDN=0,TRI=0,RST=0, VCO=50,VAS_SHDN=0,VAS_TEMP=0,CSM=0,MUTEDEL=0,CDM=0,CDIV=2, SDLDO=0,SDDIV=0,SDREF=0,BS=0,FB=1,DIVA=0,BS2=200,SDVCO=0,MTLD=0,BDIV=0,RFB_EN=0,BPWR=3,RFA_EN=1,APWR=3,
VAS_DLY=0,SDPLL=0,F01=0,LD=1,MUX_2=0,ADCS=0,ADCM=0;
Data is only transferred when LE Pin is low
Program Registers in the order 0x05->0x04->0x03->0x02->0x01->0x00
Program all registers twice upon power up with delay of 20ms in between.
To Enable Output B, RFB_EN=0;
For Fast Lock, CP=0, CDM=1;
To read Lock Detect, either use MUX Pin or Lock Detect Pin.
Output Frequency Range depends on DIVA.
BS=PhaseDetectorFrequency/50kHz.
PhaseDetectorFrequency=ReferenceFrequency*((1+DBR)/(R*(1+RDIV2))).
VCOFrequency = DesiredFrequency *DIVA.
N+F/M=VCOFrequency/PhaseDetectorFrequency.
For Small tuning of frequency, change (F/M) where F<M and M<=4095.