Project ENGN4545

- Project description
- Frequency Synthesisers
- Introduction to the Radiofrequency chips used in the projects.

► Filters



ENGN4545 Project

- Two projects. A choice that depends on your skills and preferences.
- Transceiver 1. Based on the LM1496 balanced modulator/demodulator. Discrete design. Lots of practice in applying theory learnt in the course.
- Transceiver 2. Based on integrated solutions: AD9874 (IF digitising subsystem) as the receiver and the AD9854 (Direct digital synthesiser) as the transmitter.
- Work in pairs, collaborate as a class.







- Discrete. Design each sub-circuit.
- > Easy to understand all the details and modify the design.
- Very flexible because it is a low IF down/up converter. Can handle any modulation scheme by leaving it to the baseband processor.
- Cheaper and faster in a one off.
- Easiest for those who understand the theoretical aspects of the course well and are good at RF design.
- Can do a breadboard dead-bug prototype to get working before you do the PCB. (Advisable)







- Integrated. Plug-n-play.
- ▶ Need to read the datasheets for the AD9854 and AD9874 carefully.
- Also very flexible for the same reason.
- Cheaper and faster in the long run.
- Ideal for those who wish to experience state of the art and are good at RF PCB design and soldering.
- Cannot do a breadboard dead-bug prototype. Start with a PCB layout. I.E. start EAGLE (or whatever) design immediately.



Frequency Synthesisers

- Provide the source of the Radiofrequency carrier signals
- ► Two types of circuit used...
- Phase lock loop (PLL) and voltage controlled oscillator (VCO)
- Direct Digital Synthesiser (DDS).



Phase Lock Loop and Voltage Controlled Oscillator





Phase Lock Loop and Voltage Controlled Oscillator

- Phase detector
- > Prescalers to allow frequency tuning: $Frequency = (N/R)F_{CLOCK}$
- > Thus F_{CLOCK}/R controls the frequency resolution.
- Loop filter to control the dynamic response
- Voltage controlled oscillator (VCO). For radiofrequency devices this is normally to be provided by the user as the quality output signal depend mostly on the VCO.



The MC145170

- Operates up to 185 MHz. But no internal VCO.
- ► R Counter Division Range: 1 and 5 to 32,767
- ► N Counter Division Range: 40 to 65,535
- Special patented bit grabbing interface to set the PLL parameters such as N and R prescaler values.







MC145170–2 BLOCK DIAGRAM



The Voltage Controlled Oscillator





A totally digital device that produces a sine wave!
FUNDAMENTAL DIRECT DIGITAL SYNTHESIS SYSTEM





- An address counter fed by a clock cyclically reads out addresses into a sine wave look up table.
- > The bytes in the L.U.T. represent a sine wave stored in memory.
- The L.U.T. is connected to a register that couples the L.U.T. address contents to a DAC, thus producing an anlogue sine wave.



- A fundamental problem with this architecture is that the frequency cannot be varied except by changing the clock frequency.
- Consider a system in which the delta phase or phase increment is stored as well and added to a phase accumulator.
- > This the **numerically controlled oscillator** hardware.
- The frequency at the output of the DDS is given by $f = (M f_{CLOCK})/2^M$.
- This gives a highly resolvable frequency. 48 bits of resolution is common in the L.U.T.
- However the L.U.T. readout now occurs every M adress locations, and so the DDS is subject to Nyquist's theorem.
- Thus the highest frequency is half the clock frequency!











AD9854 Direct Digital Synthesiser

FUNCTIONAL BLOCK DIAGRAM





Perspective on Frequency Synthesisers

- PLL and VCOs are the way to produce up to microwave electromagnetic waves. This is because some form of tuning element is available to vary the frequency of the VCO which may work in the microwave range.
- For frequencies above about 10 GHz special semiconductor devices e.g. Impatt sources and Gunn diodes become available for low power applications. However these still need some frequency stabilisation.
- DDS and NCOs are complete digital solutions and are therefore restricted to frequencies.
- Often we need to embed a frequency synthesiser in digital hardware. In one limit this could be a sine wave generator in MATLAB when timing is non critical.
- For time critical applications wecan use a Coordinate Rotating Digital Computer (CORDIC).



AD9874 IF Digitising Subsystem





Filters

Two different types...

- **LC ladder** Low pass (High pass).
- > LC ladder Bandpass for large bandwidth.
- Helical or LC resonator for narrow band bandpass.





Low pass design

- ► Note the component numbering scheme.
- ► Normalisation as follows...

$$C'_k = \frac{C_k}{2\pi f_{3dB} Zo} \qquad L'_k = \frac{Z_o L_k}{2\pi f_{3dB}}$$

$$Z_o = R_S = R_L$$





LOPASS.m

```
%Number of frequencies
Nvals = 200;
%Specify the number of nodes
N = 5;
%initialise all ys
y = zeros(N,N-1,Nvals);
fc= 50.e6;
Zo=50;
frequency = linspace(1.e6,1000e6,Nvals);
omega = 2*pi*frequency;
%specify an L and C in SI units
C1=1.262; L2=1.520; C3=2.239; L4=1.680; C5=2.239; L6=1.520; C7=1.262;
C1 = C1 / (2*pi*fc) /Zo
L2 = Zo*L2 / (2*pi*fc)
C3 = C3 / (2*pi*fc) /Zo
L4 = Zo*L4 / (2*pi*fc)
C5 = C5 / (2*pi*fc) /Zo
L6 = Zo*L6 / (2*pi*fc)
C7 = C7 / (2*pi*fc) /Zo
y(1,2,:) = 1/j./omega/L2; y(3,2,:) = 1/j./omega/L4; y(4,3,:) = 1/j./omega/L6;
y(2,1,:) = 1/j./omega/L2; y(2,3,:) = 1/j./omega/L4; y(3,4,:) = 1/j./omega/L6;
y(1,5,:) = j*omega*C1 + 1/Zo; y(2,5,:) = j*omega*C3; y(3,5,:) = j*omega*C5;
y(5,1,:) = j*omega*C1 + 1/Zo; y(5,2,:) = j*omega*C3; y(5,3,:) = j*omega*C5;
y(4,5,:) = j*omega*C7 + 1/Zo;
y(5,4,:) = j*omega*C7 + 1/Zo;
```



solve_LOPASS.m

```
close all
clear all
set(0, 'DefaultLineLineWidth',3)
set(0, 'DefaultAxesFontSize',20)
set(0, 'DefaultTextFontSize',20)
LOPASS;
m = zeros(N-1,N-1,Nvals);
Current_Source = [1,zeros(1,N-2)]';
<u>%</u>
mm = sum(y, 1);
for i = 1:N-1
      m(i,i,:) = mm(1,i,:);
      for j = 1:N-1
            if i~=j
                 m(i,j,:) = -y(j,i,:);
            end
      end
end
Volts = zeros(N-1,Nvals);
for i = 1:Nvals
       Volts(:,i) = inv(m(:,:,i))*Current_Source;
end
TF = 2/Zo*Volts(4,:);
TF = 20*log10(abs(TF));
semilogx(frequency,TF);
xlabel('Frequency (MHz)')
ylabel('dB')
axis([10e6 1000e6 -100 10])
title('transfer function')
```

