Université Hassan 1er
Ecole Nationale des Sciences Appliquées
Khouribga

3éme Année cycle ingénieur : Génie Electrique

Rapport des travaux pratiques

Conception analogique mixte et séminaires : VHDL-AMS

Réalisé par :

EL FAHIMI Oussama N : 16

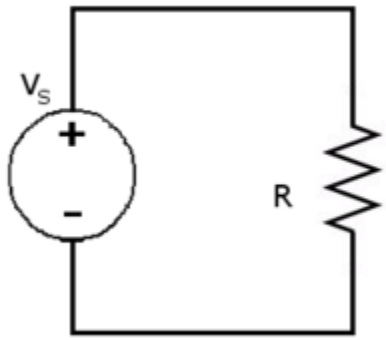
NASSABI Kanza N : 39

LARHRIB Hamza N : 34

Encadré par :

M .ISMAIL LAGRAT

On considère le circuit suivant :



On commence par définir de la source de tension sinusoïdale :

```
LIBRARY DISCIPLINES;
LIBRARY IEEE;

USE DISCIPLINES.ELECTROMAGNETIC_SYSTEM.ALL;
USE IEEE.MATH_REAL.ALL;

ENTITY Genrator IS
    PORT(TERMINAL T1,T2: ELECTRICAL);
END;


ARCHITECTURE behav OF Generator IS

    QUANTITY V_Gen ACROSS I_Gen THROUGH T1 TO T2;
BEGIN

     V_Gen==10.0 * sin (2.0*3.14*500.0 * now);
END;
```

Après la définition de la résistance :

```
LIBRARY DISCIPLINES;
USE DISCIPLINES.ELECTROMAGNETIC_SYSTEM.ALL;

ENTITY Res IS
    PORT     (TERMINAL T1,T2  : ELECTRICAL);
END Res;

ARCHITECTURE behav OF Res IS
    QUANTITY V_Res ACROSS I_Res THROUGH T1 TO T2;
BEGIN
    I_Res == V_Res/10.0;
END behav;
```

TestBench :

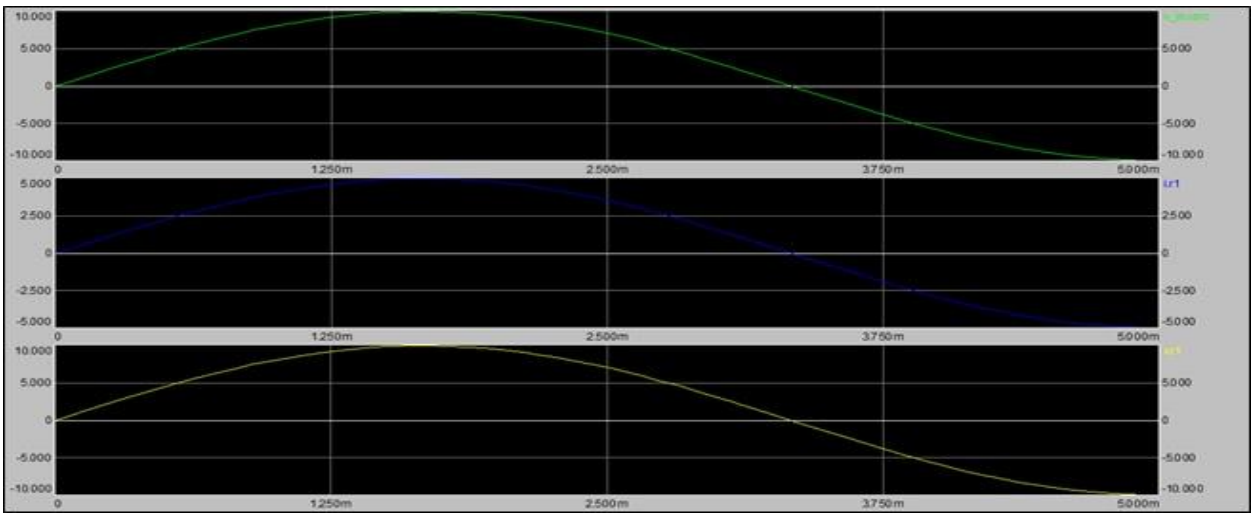
```
LIBRARY DISCIPLINES;
USE DISCIPLINES.ELECTROMAGNETIC_SYSTEM.ALL;

ENTITY network IS
END;

ARCHITECTURE behav OF network IS
    TERMINAL n1,n2: ELECTRICAL;
BEGIN
    vsrc: ENTITY Generator (behav) PORT MAP (n1, electrical_ground);
    r1:   ENTITY Resistor   (behav) PORT MAP (n1, n2);

END;
```

La simulation :



Exemple : Résistance non linéaire avec un effet du bruit :

On a tapé le code suivant :

```
LIBRARY DISCIPLINES;
LIBRARY IEEE;

USE DISCIPLINES.ELECTROMAGNETIC_SYSTEM.ALL;
USE IEEE.MATH_REAL.ALL;

entity resistor is
generic ( R : real; temp : real := 1.0; R0:real := 500.0E+3);
port (terminal t1,t2 : electrical);
end resistor;

architecture noisy of resistor is
constant ki :real :=1.0E-5;
quantity therminalNoiseSource : real noise 4.0*k*temp/R0;
quantity VR across IR through t1 to t2;
begin
IR ==VR/R+therminalNoiseSource;
end noisy
```

Exemple 2: Comparateur à hystérésis :

Le code suivant représente le modèle :

```
library IEEE, Disciplines;
use IEEE. std_ logic_ 1164.all;
use Disciplines. electrical_ system. all;
entity ComparatorHyst is
generic (vlo, vhi: REAL; -- thresholds
timeout: DELAY_ LENGTH);
port (terminal ain, ref: electrical;
signal dout: out std_ logic);
end entity ComparatorHyst;

architecture Hysteresis of ComparatorHyst is
type states is (unknown, zero, one, unstable);
quantity vin across ain to ref;
function level( vin, vlo, vhi: REAL) return states is
begin
if vin < vlo then return zero;
elsif vin > vhi then return one;
else return unknown; end if;
end function level;
begin
process
```

```
variable state: states := level( vin, vlo, vhi);
begin
case state is
when one => dout <= ’1’;
wait on vin’Above( vhi);
 state := unstable;
when zero => dout <= ’0’;
wait on vin’Above( vlo);
state := unstable;
when unknown => dout <= ’X’;
wait on vin’Above( vhi), vin’Above( vlo);
state := level( vin, vlo, vhi);
when unstable =>
 wait on vin’Above( vhi), vin’Above( vlo) for timeout;
state := level( vin, vlo, vhi);
end case;
end process;
end architecture Hysteresis;
```

Comparateur (PLL):

Le comparateur de phase est un système électronique qui a pour fonction de générer un signal de sortie proportionnel à la différence de phase entre deux signaux d'entrée.

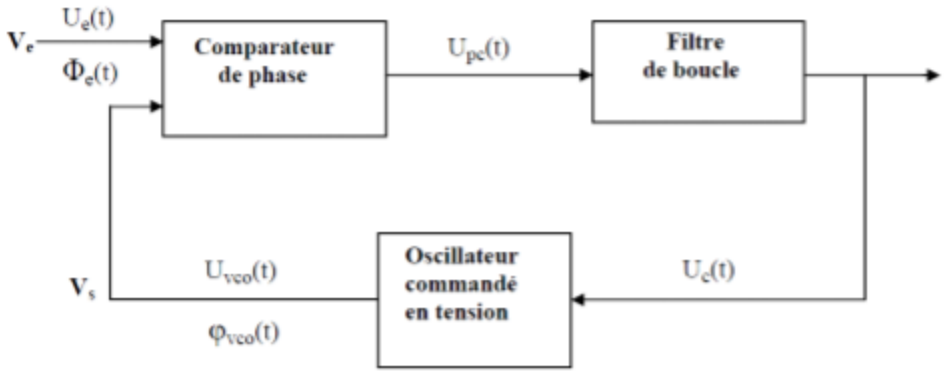
Le signal généré peut être de nature différente selon le type de comparateur :

- Une tension de sortie proportionnelle à la différence de phase entre les deux entrées.
- Un écart temporel entre deux sorties proportionnel à la différence de phase entre les deux entrées.

On distingue de plus,

- les systèmes qui détectent les écarts de phase.
- les systèmes qui détectent les écarts de phase et les écarts de fréquence.

- les systèmes qui comparent des signaux d'entrée analogiques et digitaux.
- les systèmes qui comparent des signaux digitaux en entrée seulement.



Modélisation d'un PLL :

La PLL sera donc modélisée à partir de ses fonctions de base : le multiplieur, le VCO et le filtre. Le multiplieur ou détecteur de phase est décrit par la multiplication de deux signaux d'entrée, l'un représentant le signal à démoduler, l'autre étant le signal sortant du VCO. La sortie de cette fonction est écrêtée entre –1 et +1 V. Le multiplieur est caractérisé par son facteur de conversion phase – tension Kc. Le VCO est décrit par une fonction sinusoïdale dont le terme de déphasage est contrôlé par la tension de commande. Le VCO est caractérisé par sa fréquence libre Fo et son facteur de conversion tension-fréquence Kvco. Le filtre du premier ordre est quant à lui décrit par sa Transformée de Laplace. Ce filtre, de type RC, est défini par sa constante de temps τ.

Comparateur de phase :

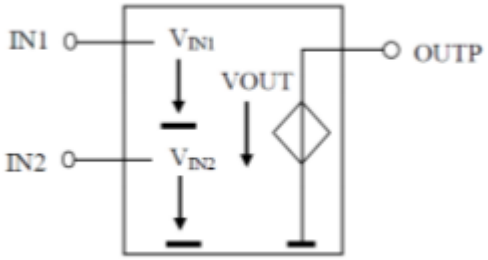


Figure 1 Structure du modèle de base du détecteur de phase

Le modèle VHDL-AMS du comparateur de phase analogique :

```
LIBRARY IEEE;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;
ENTITY phase_detect IS
GENERIC (GAIN : REAL :=1.0);
PORT(TERMINAL IN1 : ELECTRICAL;
TERMINAL IN2 : ELECTRICAL;
TERMINAL OUTP : ELECTRICAL);
END ENTITY phase_detect;
ARCHITECTURE basic OF phase_detect IS
QUANTITY Vin ACROSS IN1;
QUANTITY Vin ACROSS IN2;
QUANTITY Vout ACROSS Iout THROUGH outp;
BEGIN
VOUT == GAIN*VIN1*VIN2;
END ARCHITECTURE basic;
```

Filtre de boucle :

```
LIBRARY IEEE;
LIBRARY DISCIPLINES;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.MATH_REAL.ALL;
USE DISCIPLINES.ELCTROMAGNETIC_SYSTEMS.ALL;
ENTITY filter IS
GENERIC(GAIN: REAL := 1.0;
FC: REAL := 1.0e6;
ZF: REAL := 1.0e6);
PORT(TERMINAL inp : ELECTRICAL;
TERMINAL outf : ELECTRICAL);
END ENTITY filtre;
ARCHITECTURE behav OF filter IS
QUANTITY Vin ACROSS inp;
QUANTITY Vout ACROSS iout THROUGH outf;
constant num;
real_vector(1 to 2):= (10.0,0.0);
constant den;
real_vector(1 to 2):= (1.0,1.0/1.0);
BEGIN
Vout== Vin'LTF(num,den);
END ARCHITECTURE behav;
```

Oscillateur commandé en tension :

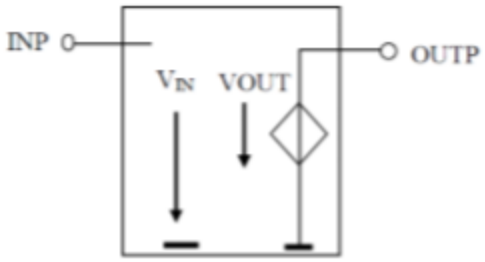


Figure 2 Structure du modèle de base du VCO

Un oscillateur analogique commandé en tension (VCO) génère un signal sinusoïdale dont la fréquence est proportionnelle à la valeur d'une tension de commande.

```
LIBRARY IEEE;
USE IEEE.MATH_REAL.ALL;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;
ENTITY vco IS
GENERIC (GAIN : REAL := 1.0; -- gain du vco [-]
KVCO: REAL := 10.0e6; -- pente (sensibilite) du vco [HZ/V]
FC : REAL := 1.0E3; -- frequence centrale [HZ]
UC0 : REAL := 1.2 -- tension de commande a F=FC [V]
);
PORT (TERMINAL INP, OUTP: ELECTRICAL);
END ENTITY vco;
ARCHITECTURE behav OF VCO IS
CONSTANT WC : REAL := MATH_2_PI * FC;
CONSTANT KVC : REAL := MATH_2_PI * KVCO; -- [RAD/(S*V)]
QUANTITY Uin ACROSS INP TO GROUND;
QUANTITY OUTP ACROSS Iout THROUGH OUTP TO GROUND;
QUANTITY DP, P: REAL; -- PHASE
BEGIN
DP == WC + KVC *(UIN - UC0);
P == DP'INTEG;
UOUT == GAIN * SIN(P);
END ARCHITECTURE behav;
```

Test Bench :

On se basant sur le schéma fonctionnel du PLL, la consigne du PLL est un signal sins de 3Mhz .

```
LIBRARY DISCIPLINES;
USE DISCIPLINES.ELECTROMAGNETIC_SYSTEM.ALL;

ENTITY tb IS
END;

ARCHITECTURE behav OF tb IS
    TERMINAL in1,outc,  outf, outp : ELECTRICAL;
BEGIN
    v_in1: ENTITY sineSource (behav) PORT MAP (in1, electrical_ground);
    comp:   ENTITY phase_detect    (basic) PORT MAP (in1, outp,outc);
    filtre:   ENTITY filtre          (behav) PORT MAP (outc, outf);
    vco:   ENTITY vco            (behav) PORT MAP (outf, outp);
END;
```

La simulation :