

Corrigé TD Electronique numérique : exercices ABEL 1, 2, 3, 4.
Description en langage ABEL

1. Un multiplexeur 4->1.

```

MODULE exo1
TITLE 'multiplexeur'
a3..a0 pin 3..6 ;           // 4 entrées
c1,c0 pin 11,10;          // contrôle multiplexeur
y      pin 12 istype 'com' ; // sortie combinatoire
select=[c1,c0] ;          // « set »

EQUATIONS
y = a0 & (!c1 & !c0)
  # a1 & (!c1 & c0)
  # a2 & ( c1 & !c0)
  # a3 & ( c1 & c0) ;
END

```

ou avec une description de plus haut niveau :

```

...
WHEN (select==0) THEN y=a0
  ELSE WHEN (select==1) THEN y=a1
    ELSE WHEN (select==2) THEN y=a2.
      ELSE y=a3;
END ;

```

2. Un compteur 2 bits.

```

MODULE exo2
TITLE 'compteur avec raz'
Horloge pin 2 ;
Raz      pin 3 ;           // remise à zéro
Q1,Q0 pin 12,13 istype 'reg' ; // sorties

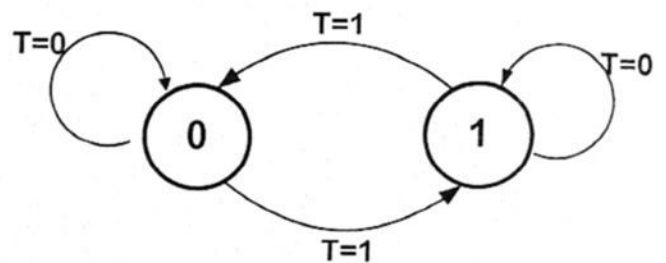
compteur=[c1,c0] ;        // « set »

EQUATIONS
compteur.clk=horloge;
compteur.clr=Raz ;       // reset synchrone
compteur:=compteur+1;
END

```

3. On désire réaliser une bascule T à partir d'une bascule D.

Dessinez le graphe de transition de la bascule T :



Traduire en langage ABEL ce graphe en utilisant
3.1 La description STATE_DIAGRAM

```

MODULE exo3
TITLE 'compteur avec raz'
Horloge pin 2 ;
T      pin 3 ; // Entrée T
Q0     pin 12 istype 'reg' ; // sortie

S0=0 ; S1=1 ; // états symbolique

EQUATIONS
Q0.clk=Horloge;

STATE_DIAGRAM Q0
STATE S0: IF T THEN S1;
           ELSE S0;
STATE S1: IF T THEN S0;
           ELSE S1;
END

```

3.2 La description Table de vérité TRUTH_TABLE

```
...
EQUATIONS
Q0.clk=Horloge;
TRUTH_TABLE ([ Q0 , T ] :> [Q0])
    [ 0 , 0 ] :> 0 ;
    [ 0 , 1 ] :> 1 ;
    [ 1 , 0 ] :> 1 ;
    [ 1 , 1 ] :> 0 ;
END
```

3.3 La description par équation Booléenne EQUATIONS

```
...
EQUATIONS
Q0.clk=Horloge;
Q0:= Q0 $ T;
END
```

4. Concevoir un chenillard à motifs en utilisant l'opérateur de décalage circulaire 4bits (*TPI*) et un compteur 2 bits pour fournir le code de commande C1, C0. (*Exercice du TP3*).

```
MODULE TPexol
TITLE 'chenillard'

a3..a0 pin 3..6 ; // entrees motif
s3..s0 pin 17..20 istype 'com'; // sorties
c1,c0 node istype 'reg';
horloge pin 2;

compt=[c1,c0];
sorties=[s3..s0];
entrees=[a3..a0];

EQUATIONS
compt.clk=horloge;
compt:=compt+1;

sorties= (compt==0)& entrees
#(compt==1)&[a2,a1,a0,a3]
#(compt==2)&[a1,a0,a3,a2]
#(compt==3)&[a0,a3,a2,a1];

TEST VECTORS
([horloge,entrees]->[sorties])
@repeat 4{ [.C.,5]->[.X.];}
@repeat 4{ [.C.,1]->[.X.];}

END
```