

# MI41 : Compte-Rendu de Travaux Pratiques N°4

## Calculatrice

Pierre Mauduit  
Cedric Le-Breton

16 décembre 2006

# 1 Introduction

Durant cette quatrième séance de travaux pratiques, le but est double :

- Etudier une unité entière simplifiée
- Mettre en oeuvre un graphe de commande

Pour cela, nous allons réaliser une calculatrice, dont les capacités seront limitées à l'opération +.

# 2 Le décodeur d'instruction

Le premier composant à concevoir dans cette séance de travaux pratiques est le décodeur d'instruction de la calculatrice. Donnons les entrées et les sorties mises en jeu ici :

En entrée :

- Cmd

En sortie :

- Sel
- Sel\_E
- Sel\_A
- Sel\_B
- nload
- Cu

Le code VHDL correspondant à ce composant est donné ci-dessous :

```
library ieee ;
// Libs utilisées
use ieee.std_logic_1164.all ;
use ieee.std_logic_unsigned.all ;
use ieee.std_logic_arith.all ;

entity instr_decode is
port(
    Cmd : in std_logic_vector(5 downto 0); //mot en entrée
    sel ,nload ,sel_A ,sel_B ,sel_E : out std_logic ;
    Cu : out std_logic_vector(1 downto 0)
);

end instr_decode ;
```

Les variables évoluant sans arrêt, nous avons du utiliser un signal dans le but de leur donner une valeur.

Sel correspond au premier bit du mot d'entrée, cela permet de savoir sur quel registre stocker le resultat.

Sel\_B correspond au deuxième bit car il s'agit du registre source 1 et Sel\_A correspond au troisième bit car il s'agit du registre source 2.

Sel\_B et Sel\_A correspondent respectivement au deuxième et au troisième bit d'entrée, sélectionnant le registre source 1 et source 2.

Cop (code opérateur) permet de déterminer la valeur de Cu et nload suivant les cas.

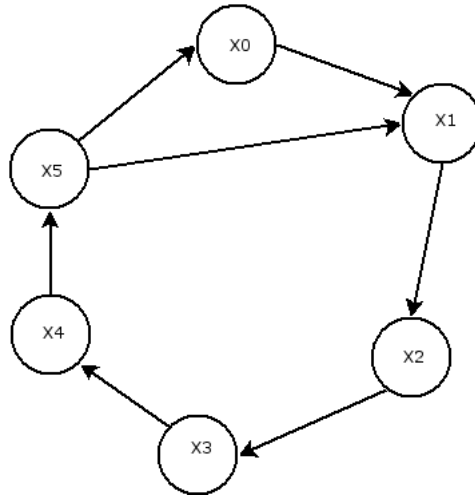
```
architecture a of instr_decode is
    signal cop : std_logic_vector(2 downto 0);
    signal s1,s2,d : std_logic;
begin
    process (Cmd)
    begin
        cop <= Cmd(5 downto 3) ;
        s1 <= Cmd(2) ;
        s2 <= Cmd(1) ;
        d <= Cmd(0) ;

        sel <= d ;
        sel_A <= s1 ;
        sel_B <= s2 ;
        sel_E <= '1' ;

        case cop is
            when "000" => Cu <= "00" ; nload <= '0' ;
            when "001" => Cu <= "01" ; nload <= '0' ;
            when "010" => Cu <= "10" ; nload <= '0' ;
            when "011" => Cu <= "11" ; nload <= '0' ;
            when "100" => Cu <= "10" ; nload <= '1' ;
            when "101" => sel_E <= '0' ; nload <= '0' ;
            when others => Cu <= "00" ; nload <= '1' ;
        end case ;
    end process ;
end a ;
```

### 3 L'Unité de commandes simplifiées

Donnons tout d'abord le graphe à état de l'addition que l'on cherche à implémenter dans la calculatrice :



Les différents états représentés signifient :

- X0 : affichage
- X1 : chargement 1re valeur
- X2 : changement de registre
- X3 : attend l'opérateur
- X4 : chargement deuxième valeur et sauvegarde de l'opérateur
- X5 : choix opération suivante

Le Code VHDL correspondant est donné ci-dessous :

```
library ieee ;
use ieee.std_logic_1164.all ;

entity simple_command_unit is
port(
    C , e , p : in std_logic;
    S : out std_logic_vector(5 downto 0);
);
end simple_command_unit;

architecture a of simple_command_unit is
type state_t is ( X0 , X1 , X2 , X3 , X4 , X5);
signal etat : state_t;
begin
```

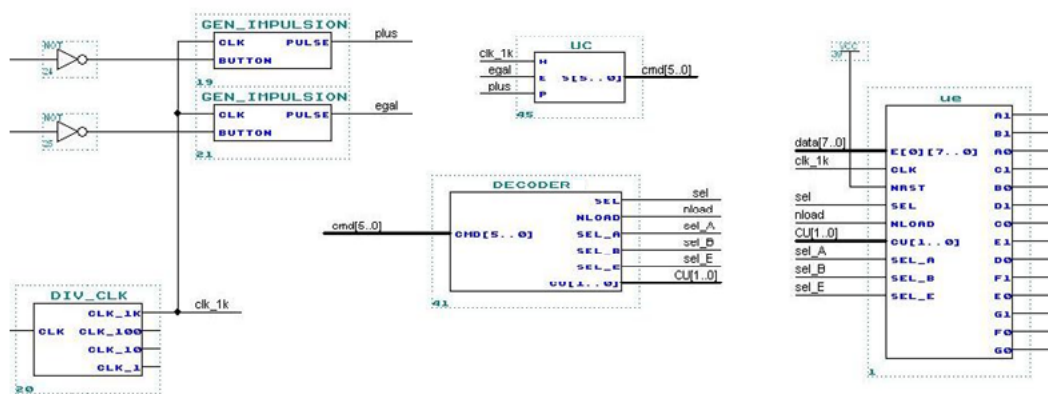
```

process(C)
variable v : std_logic ;
begin
    if( C'event and C='1' ) then
        case etat is
            when X0 => if( p='1' ) then
                etat <= X1;
            end if;
            when X1 => etat <= X2;
            when X3 => if( e='1' ) then
                etat <= X4;
                v := '0';
            else
                if (p='1') then
                    etat <= X4;
                    v := '1';
                end if;
            end if;
            when X4 => etat <= X5;
            when X2 => etat <= X3;
            when X5 => if( v='0' ) then
                etat <= X0;
            else etat <= X2;
            end if;
        end case;
    end if;
end process;

S <= "101001" when etat = X1 else
    "101000" when etat = X4 else
    "001010" when etat = X5 else
    "000001" when etat = X2 else
    "000000" ;
end a ;

```

Tous les composants étant réunis, il ne reste plus qu'à les relier entre eux, en mode graphique sous le logiciel MaxPlusII.



## 4 Conclusion

Cette séance de travaux pratiques a permis de mettre en valeur des aspects plus poussés du langage VHDL que les aspects de base vus dans les séances précédentes, en réalisant une calculatrice, objet qui peut paraître banal, mais qui est finalement loin d'être trivial, conceptuellement parlant.