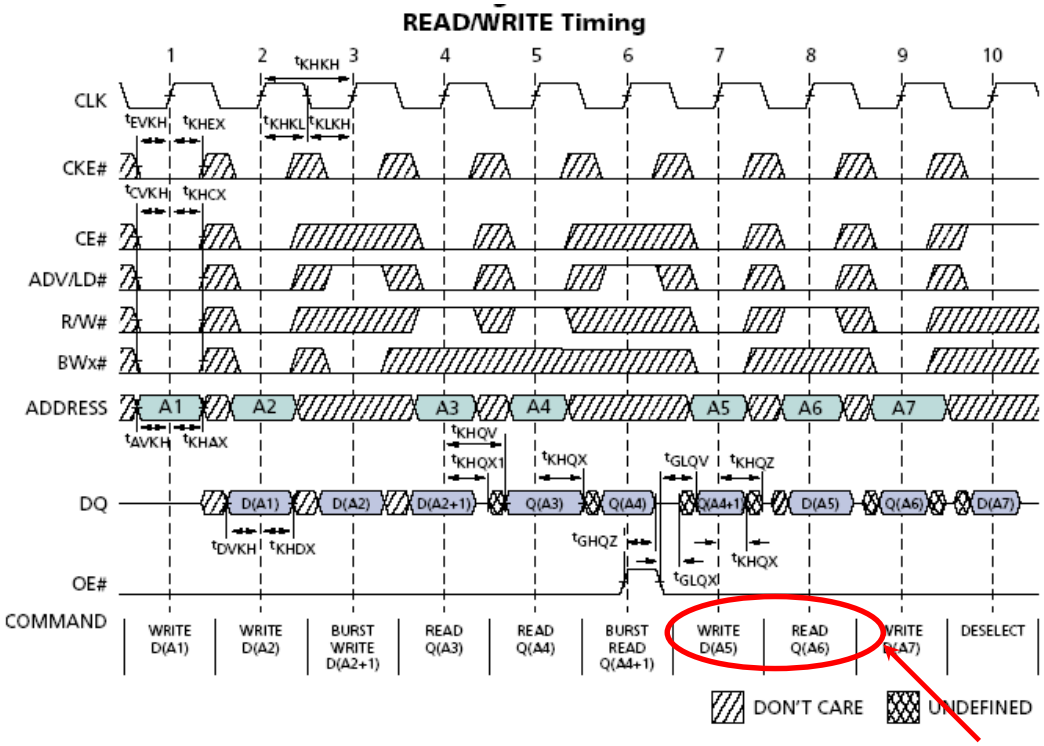
Département Informatique Electronique

3ème année du cycle d'ingénieurs ITSE51- SoC

Enseignant: Julien Dubois

TP : Description and validation of a ZBT SRAM controler

L'objectif de ce projet est de décrire un contrôleur de mémoire SRAM. Le contrôleur devra permettre de piloter une mémoire CYPRESS mt55l512y36f. Vous pourrez vous appuyez sur la documentation de cette mémoire et vous utiliserez son modèle VHDL (mt55l512y36f.vhd) fournie par le fabricant. Ce type de mémoire statique porte l'extension ZBT, cela signifie qu'il n'y a pas de perte de cycle entre une phase de lecture et d'écriture (Cf figure 1).



NOTE:
1. For these waveforms, ZZ is tied LOW.
2. Burst sequence order is determined by MODE (0 = linear, 1 = interleaved). BURST operations are optional.
3. CE# represents three signals. When CE# = 0, it represents CE# = 0, CE2# = 0, CE2 = 1.
4. Data coherency is provided for all possible operations. If a READ is initiated, the most current data is used. The most recent data may be from the input data register.

Pas de perte de cycle entre lecture et écriture

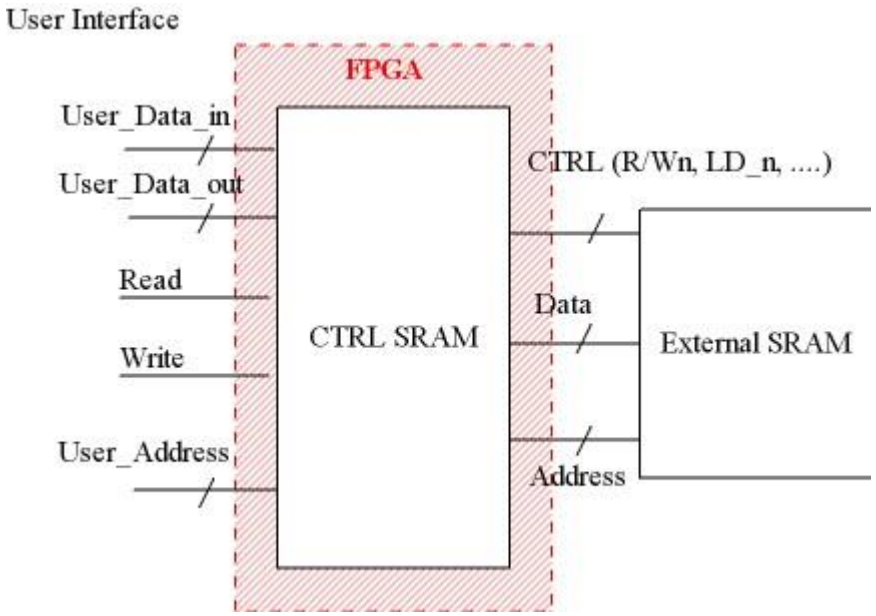
Figure 1

Vous devrez gérer les signaux de contrôle CKE#, CE#, OE#, R/W#, (ADV/LD# = LD_n), CE2#, CE2, les bus d'adresses et de données (appuyer fortement sur le diagramme de la figure 1). Attention le bus de données est bidirectionnel, il devra donc être géré pour cela avec des buffers 3 états (composant IOBUF cf Annexe 1).

Travail à réaliser :

Vous modéliserez le contrôleur **en VHDL** sans vous préoccupez des modes BURST et vous effectuerez le testbench associé. La génération d'adresse sera effectuée à l'extérieur du

composant. Le contrôleur possédera en outre obligatoirement deux signaux d'entrée READ et WRITE. Ces signaux déclencheront respectivement des phases de lecture ou d'écriture dans la mémoire SRAM. Le signal READ sera prioritaire sur le signal WRITE.



Question subsidiaire : Modifier le composant pour réaliser une suite d'écriture ou de lecture à partir d'une adresse de départ.

Conseil rendre les signaux suivant constants de la manière suivante

nBWA = '0'
nBWB = '0'
nBWC = '0'
nBWD = '0'
ZZ = '0'
MODE = Lbo_n = '0'
nCKE = '0'

Dans la suite, fixée les contraintes de temps. Quelle est la fréquence maximale que votre contrôleur peut atteindre ? Optimisez.

Annexe 1 :

Pour la description VHDL utiliser des primitives Xilinx, utiliser la librairie unisim

```
library IEEE;
use IEEE.std_logic_1164.all;
-- synopsys translate_off
        library unisim;
        use unisim.vcomponents.all;
-- synopsys translate_on
```

Utiliser les composants IOBUF presents dans la librairie unisim pour utiliser des buffers tristate

```
component IOBUF
   port (I : in std_logic;
       IO: inout std_logic;
       T : in std_logic;
       O : out std_logic);
end component;
```