Le code du simulateur est situé dans le fichier netlist_simulator.ml. De manière générale, l'orientation dans les fichiers est la même que dans le sujet du TP.
Pour lancer le simulateur, le compiler avec :
> ocamlbuild netlist_simulator.byte
puis l'utiliser avec :
> ./netlist_simulator.byte <*filename*>

Le simulateur fonctionne à partir de deux environnement qui stockent les valeurs de toutes les variables aux instants $t-1$ et $t$.
A l'instant $t$, on demande les valeurs d'entrée à l'utilisateur à l'aide du module Scanf, puis on calcule successivement toutes les valeurs des variables à l'aide de la liste triée d'équations donnée par scheduler.ml.
Les références sont récupérées dans l'environnement $t-1$. Pour éviter de stocker tous les environnement, on stocke les deux derniers en leur associant à chacun la parité du $t$ auquel ils correspondent.

La ROM et la RAM sont modélisées par deux tableaux à taille fixe, initialisés respectivement à partir du fichier binaire rom.bin (dont la validité n'est pas testée) et à 0 pour toutes les valeurs.
On cherche au préalable la taille des mots de la ROM/RAM et la longueur de la RAM pour les initialiser.
La RAM est modifiée après calcul de toutes les valeurs d'un cycle par le simulateur.

Le nombre de cycle est précisé au début, à modifier par l'utilisateur.

Les différentes fonctions print qui abondent dans le code ont servi notamment à des fins de vérifications de celui-ci.

J'ai essayé de maximiser le nombre de fonctions auxiliaires définies hors du programme principal, afin de rendre le code plus compréhensible.