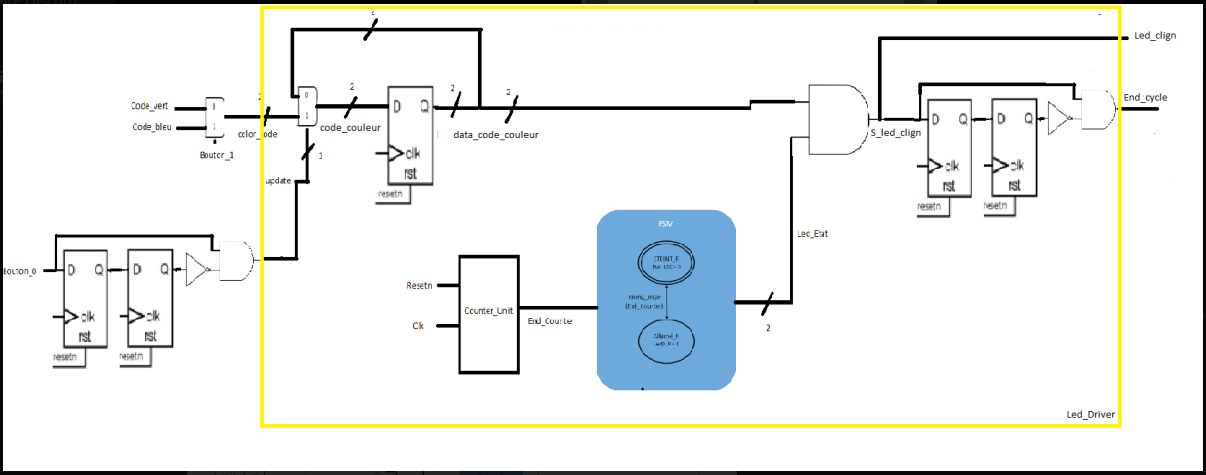
TP04 - Pilotage de LED et mémoire (Part2) (Kamal KHERCHOUCH)

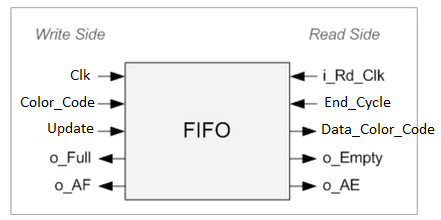
1. **Sur l’architecture RTL, modifiez le module *LED\_driver* en ajoutant une sortie *end\_cycle*. Cette sortie vaudra 1 à la fin d’un cycle allumé/éteint de la LED RGB.**



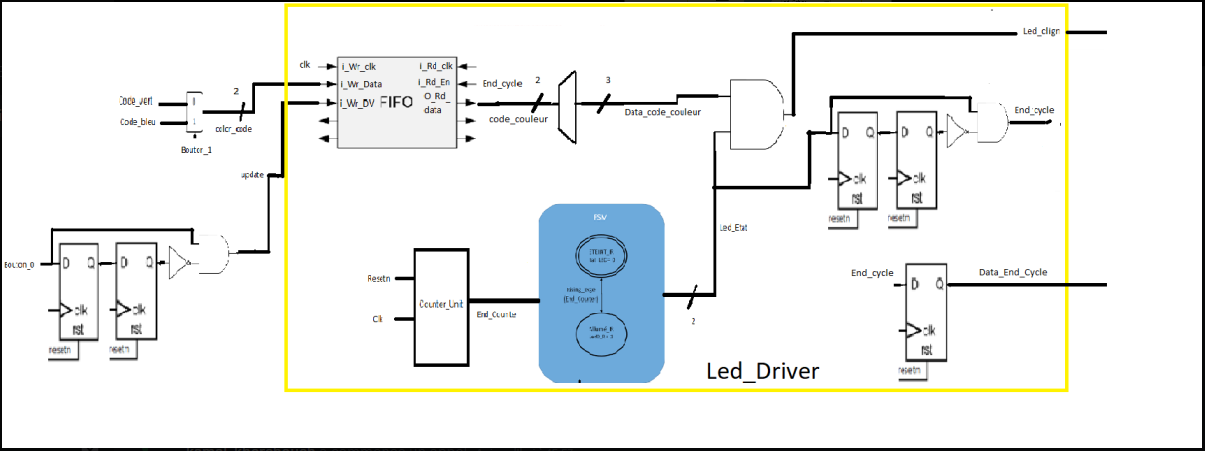
Sur l’architecture RTL de la fin du TP4 (part1), On a modifié le module Led\_Driver afin d’ajouter une détection de front montant sur le signal Led\_Clignotement. Ce qui nous permettra d’obtenir le signal end\_cycle à chaque cycle allumé/éteint de la led RGB.

1. **Modifiez la logique en entrée du module pour ajouter une FIFO. Cette FIFO doit prendre en entrée le code couleur « vert » ou « bleu » suivant l’état du bouton\_1 et est connectée en sortie à l’entrée *color\_code* du module *LED\_driver*. La donnée est écrite dans la FIFO lorsqu’il y a un front montant du bouton\_0. La donnée de la FIFO est lue lorsque le signal *end\_cycle* du module LED\_driver vaut 1.**

On ajoute, à notre schéma RTL la fifo suivante :



En remplaçant la partie combinatoire d’entrée par une Fifo, on obtient le schéma RTL suivant :



La Fifo créée un décalage entre l’entrée End\_Cycle et la sortie Code\_Couleur. Pour régler ce problème, il faut ajouter un registre sur le signal End\_Cycle afin d’avoir les signaux Led\_Clignotement et Data\_End\_Cycle synchronisé.

1. **Modifiez vos codes de la partie 1 pour y ajouter les nouveaux éléments de votre architecture.**

1ère étape : on modifie notre module Led\_Driver en y ajoutant la Fifo.

Dans l’IP catalogue on choisit le composant FIFO Generator :

