Travaux dirigés de Logique Programmable

Séance 2

L3 : 2020-2021

Rémi Griot

# Introduction

## Fichiers associés

Pour réaliser le TD, vous aurez besoin des documents suivants :

Les fichiers sources

* spimaster.vhd

Les test benchs associés :

* spimaster\_tb.vhd

## Résumé du TD2

Ce TD va vous permettre de revoir des notions du cours ainsi que de décrire une machine à état en VHDL

[1 Introduction 2](#_Toc55560896)

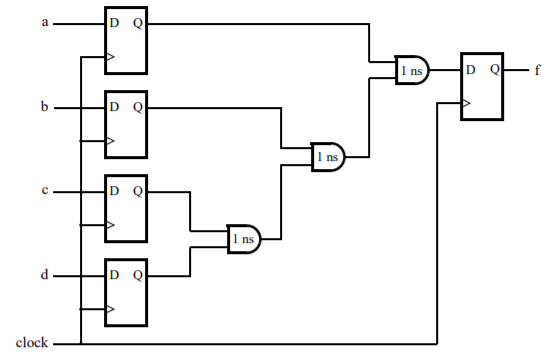
[2 Fréquence maximum d’un circuit logique 3](#_Toc55560897)

[3 Synchrone – Asynchrone / séquentiel - combinatoire 3](#_Toc55560898)

[4 Machine d’état SPI Master 4](#_Toc55560899)

# Fréquence maximum d’un circuit logique

* Calculer la fréquence maximum du circuit ci-dessous, sachant que l’on considère le skew comme négligeable, Tsu à 1ns et Tco à 1ns



* Proposer un circuit équivalent fonctionnant à une fréquence maximum plus élevée. Quelle est cette fréquence ?
* Proposer une méthode qui, en modifiant le comportement du circuit mais en gardant l’équation logique, permettrai d’autoriser une fréquence maximum plus élevé.

# Synchrone – Asynchrone / Séquentiel - Combinatoire

* Qualifier les circuits décrit ci-dessous. Dessiner le schéma associé quand il s’agit de description VHDL, et écrire la description en VHDL quand il s’agit de schéma

sig **<=** **not** sig **when** sel **=** '1' **else**

'0'**;**

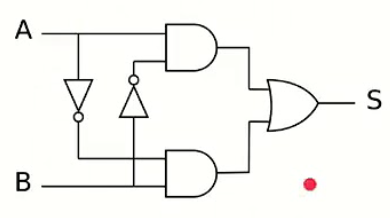
2)

sig **<=** **not** sig **when** rising\_edge (clk) ;

3)

s **<=** a **or** b **or** c**;**

4)



5)

**process** **(**clk**,** rst**)**

**begin**

**if** rst **=** '1' **then**

dd **<=** '0'**;**

ddd **<=** '0'**;**

**elsif** **rising\_edge** **(**clk**)** **then**

d **<=** a + b**;**

dd **<=** d**;**

ddd **<=** dd**;**

**end** **if;**

**end** **process;**

# Machine d’état SPI Master

Nous allons étudier et décrire une machine à état synchrone sur CLK qui constitue un sous bloc d’un ensemble servant à transférer des données sur un bus SPI. Cette machine se compose de 4 états :

**Idle – Start – Data – Stop**

Les sorties de cette machine sont

* **CS** : Chip select (~Slave Select)
* **MOSI** : Master Output Slave Input

Les entrées de cette machine sont

* **CLK** : Horloge système
* **SEND** : Signal d’envoi de données
* **TICK\_SCLK** : Signal de synchro pour décaler le registre
* **DATA2SEND**: Donnée sur 8 bit transféré
* Ecrire l’entitée de la machine d’état (spimaster)

**Idle\_state** : Etat de repos, la machine est dans cet état en sortie de reset. MOSI est à 0 et CS à 1. Passe à Start si Send est à 1

**Start\_state** : Démarrage de la communication. CS est à 0. Passe sans condition à Data.

**Data\_state** : Décale le registre contenant la Data à envoyer (DATA2SEND) sur niveau haut de TICK\_SCLK. Passe à Stop quand les 8 bit du registre ont été décalé

**Stop\_state** : Arrêt de la communication. CS passe à 1.

* Dessiner l’évolution des états
* En vous basant sur spimaster.vhd, décrire en vhdl la machine d’état
* Vérifier son fonctionnement avec spimaster\_tb.vhd