

Laboratoire 6: Conception d'une table tournante

Auteurs : *Miguel Jalube et Bastien Pillonel*

Sommaire

- [Introduction](#)
- [Conception de l'interface](#)
 - [Plan d'adressage](#)
 - [Canal d'écriture](#)
 - [Canal de lecture](#)
 - [Générateur de top](#)
 - [Diviseur de fréquence](#)
 - [Masquage des IRQ et commande moteur](#)
 - [Machines d'état](#)
 - [Machine d'état pour la calibration/initialisation](#)
 - [Machine d'état pour le déplacement](#)
 - [Machine d'état pour les IRQ](#)
 - [Synthèse](#)
- [Conclusion](#)

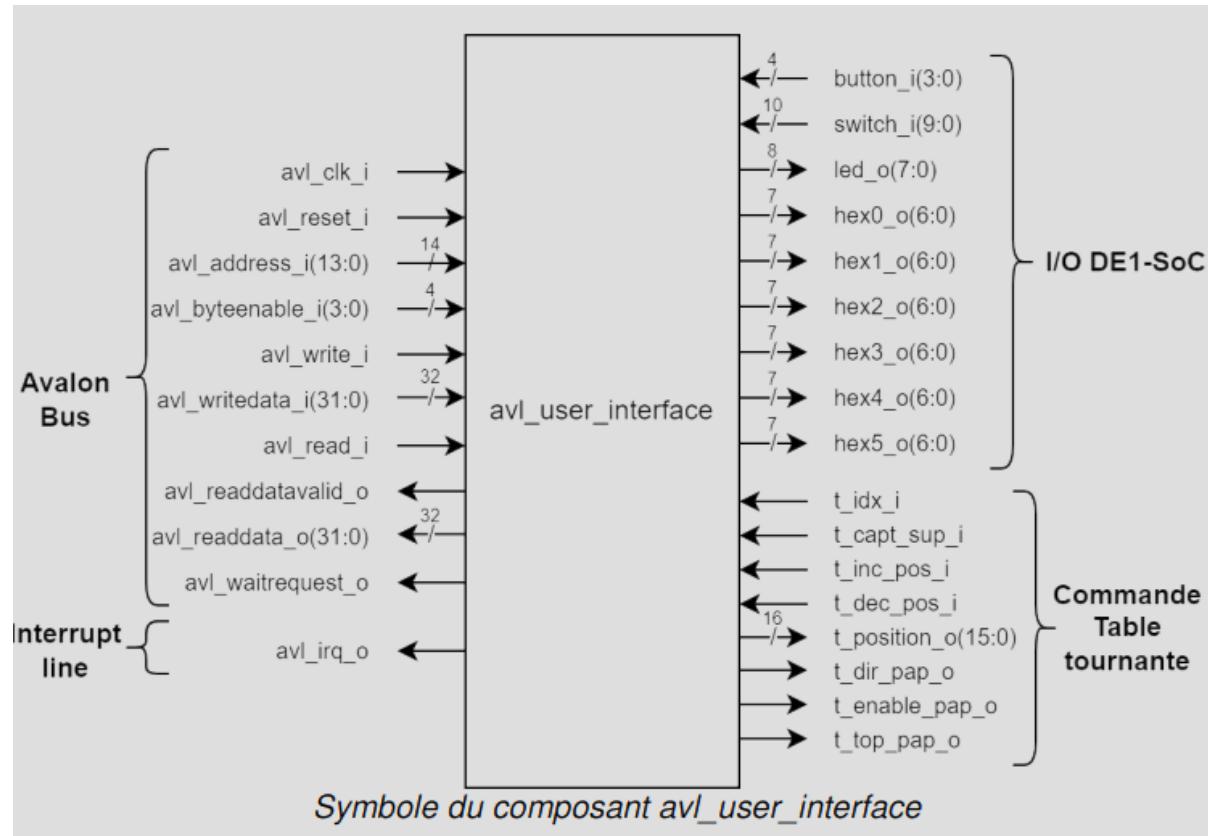
Introduction

Ce laboratoire a pour but de concevoir une application de commande d'une table tournante avec l'utilisation de plusieurs afficheurs externe. Le laboratoire comprendra plusieurs étapes :

- Mise en œuvre d'une interface UART du HPS, afin d'envoyer des messages dans un terminal distant.
- Développement et interfaçage des commandes de la table tournante.
- Développement et interfaçage des capteurs de la table tournante avec compteur de position.
- Implémentation d'un générateur de top pour le moteur afin de pouvoir sélectionner 4 vitesses différentes dans la FPGA.
- Mise en œuvre des interruptions du HPS générées par un dépassement de position de la table depuis la partie FPGA.

Conception de l'interface

Schema bloc



Plan d'adressage

Le plan d'adressage est le suivant :

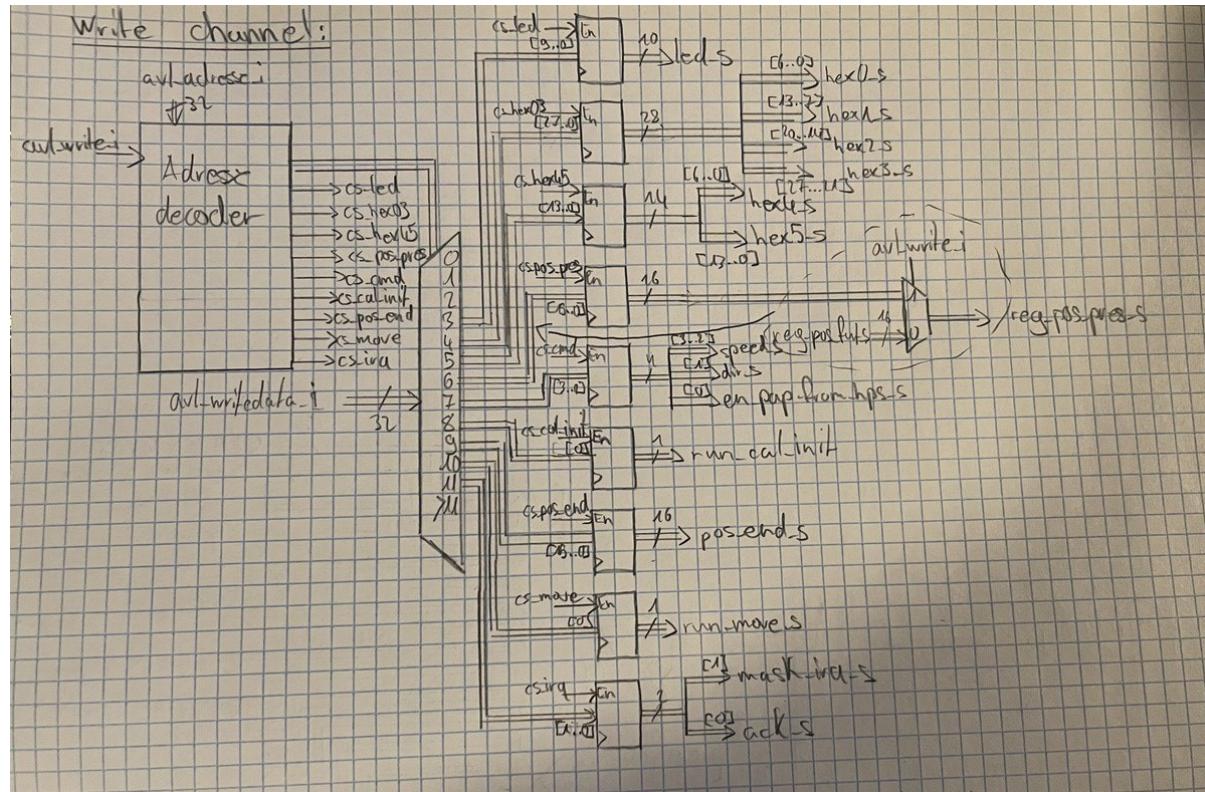
Plan d'adressage		
Adresse (offset)	READ	WRITE
0x00	[31..0] Constante ID interface	not used
0x04	[31..4] "0..0" ; [3..0] buttons	not used
0x08	[31..10] "0..0" ; [9..0] switchs	not used
0x0C	[31..8] "0..0" ; [7..0] leds	[31..8] reserved ; [7..0] leds
0x10	[31..28] "0..0" ; [27-21] hex3 ; [20-14] hex2 ; [13-7] hex1 ; [6-0] hex0	[31..28] reserved ; [27-21] hex3 ; [20-14] hex2 ; [13-7] hex1 ; [6-0] hex0
0x14	[31..14] "0..0" ; [13-7] hex5 ; [6-0] hex4	[31..14] reserved ; [13-7] hex5 ; [6-0] hex4
0x18	[31..16] "0..0" ; [15-0] val_pos;	[31..16] reserved ; [15-0] val_pos;
0x1C	[31..4] "0..0" ; [3-2] speed_s; [1] dir_s; [0] en_pap_s	[31..4] reserved ; [3-2] speed_s; [1] dir_s; [0] en_pap_s
0x20	[31..1] "0..0" ; [0] cal_busy_s;	[31..1] reserved; [0] run_cal_init_s;
0x24	[31..16] "0..0" ; [15-0] pos_end_s;	[31..16] reserved ; [15-0] pos_end_s;
0x28	[31..1] "0..0" ; [0] move_busy_s;	[31..1] reserved; [0] move_busy_s;
0x2C	[31..2] "0..0" ; [1] limit_max_s; [0] limit_min_s;	[31..2] reserved ; [1] mask_irq_s; [0] ack_s;

Les adresses disponibles pour les nouvelles fonctionnalités ont été utilisées comme suit :

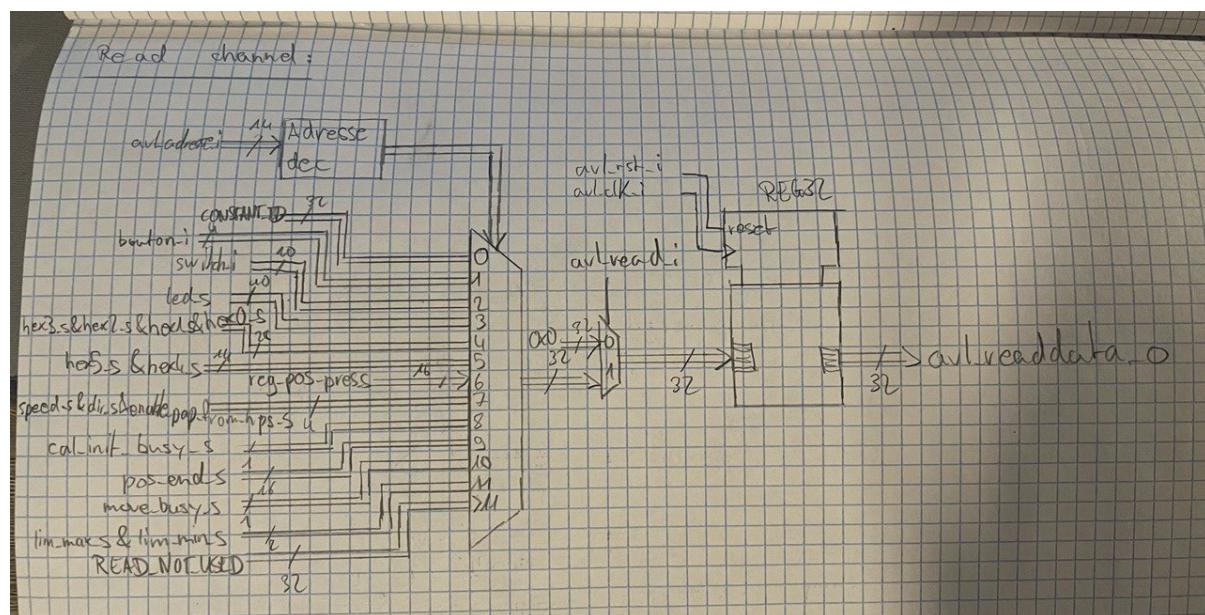
- 0x18 : lecture et écriture -> position courante
- 0x1c : lecture et écriture -> bits 3..2 : vitesse moteur, bit 1 : direction moteur, bit 0 : enable moteur

- 0x20 : lecture -> bit 0 : calibration/initialisation en cours, écriture -> bit 0 : lancer calibration/initilisation
- 0x24 : lecture et écriture -> bit 0 : position après depl. auto
- 0x28 : lecture et écriture -> déplacement en cours
- 0x2c : lecture -> bit 1 : limite maximale atteinte, bit 0 :limite minimale atteinte, écriture -> bit 1 : masquage des irq, bit 0 : acknowledge irq

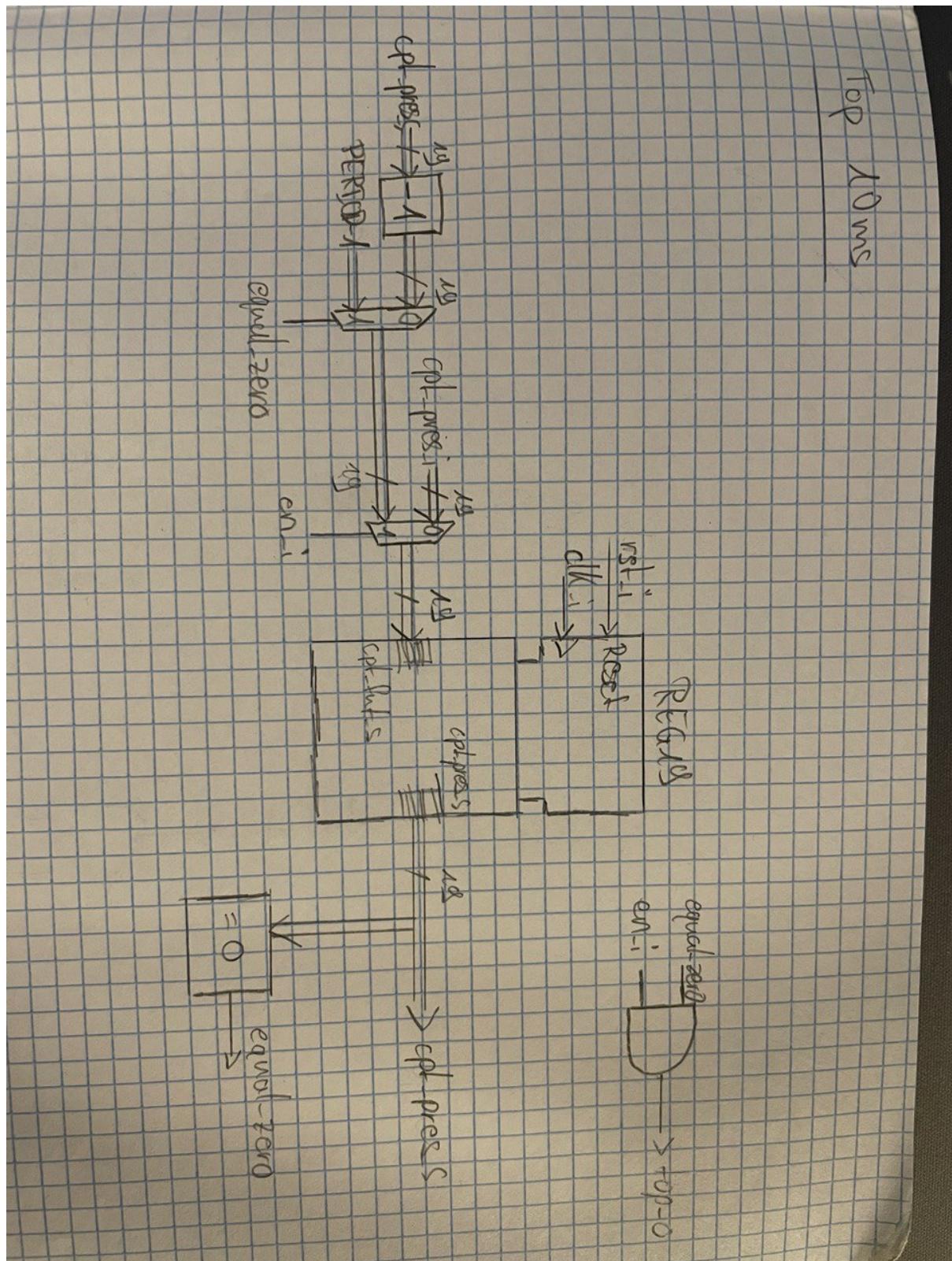
Canal d'écriture



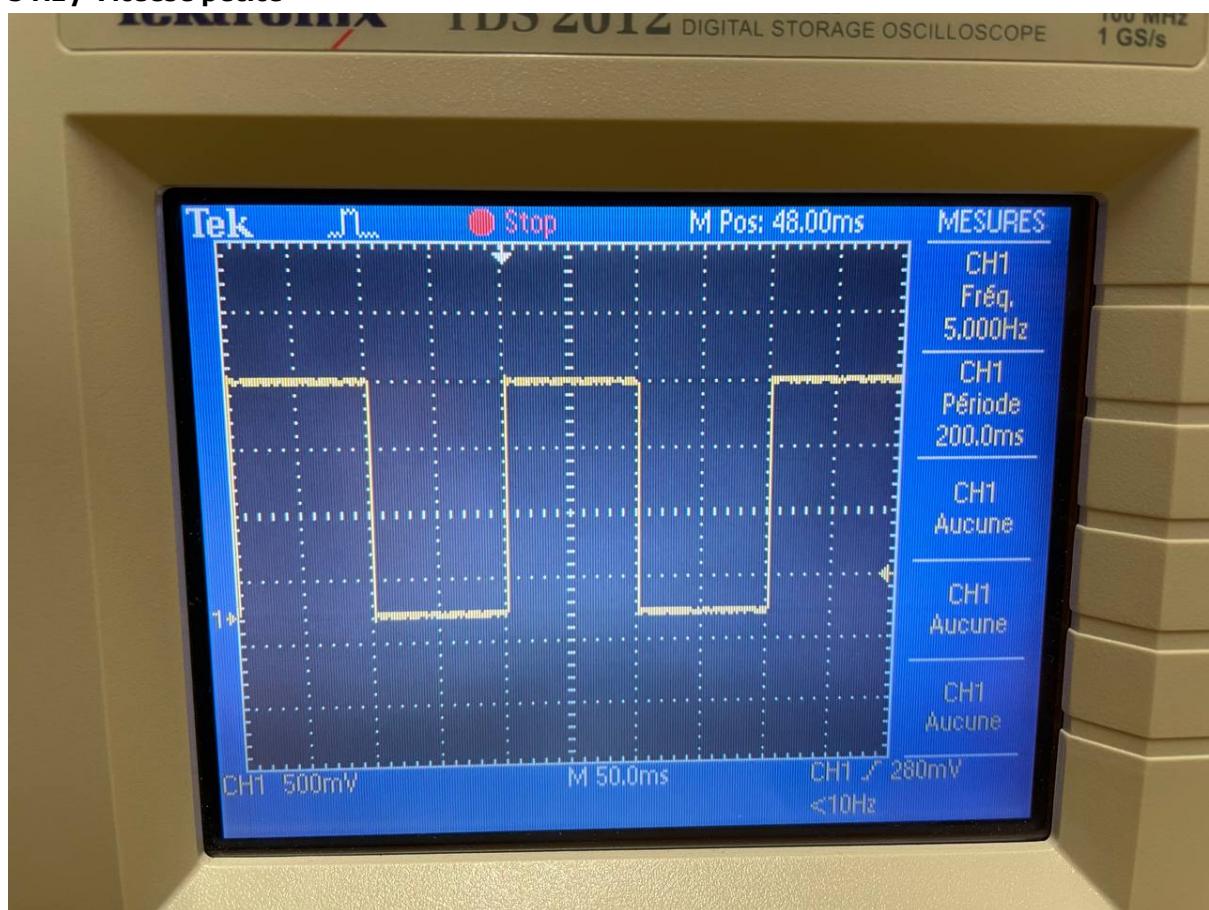
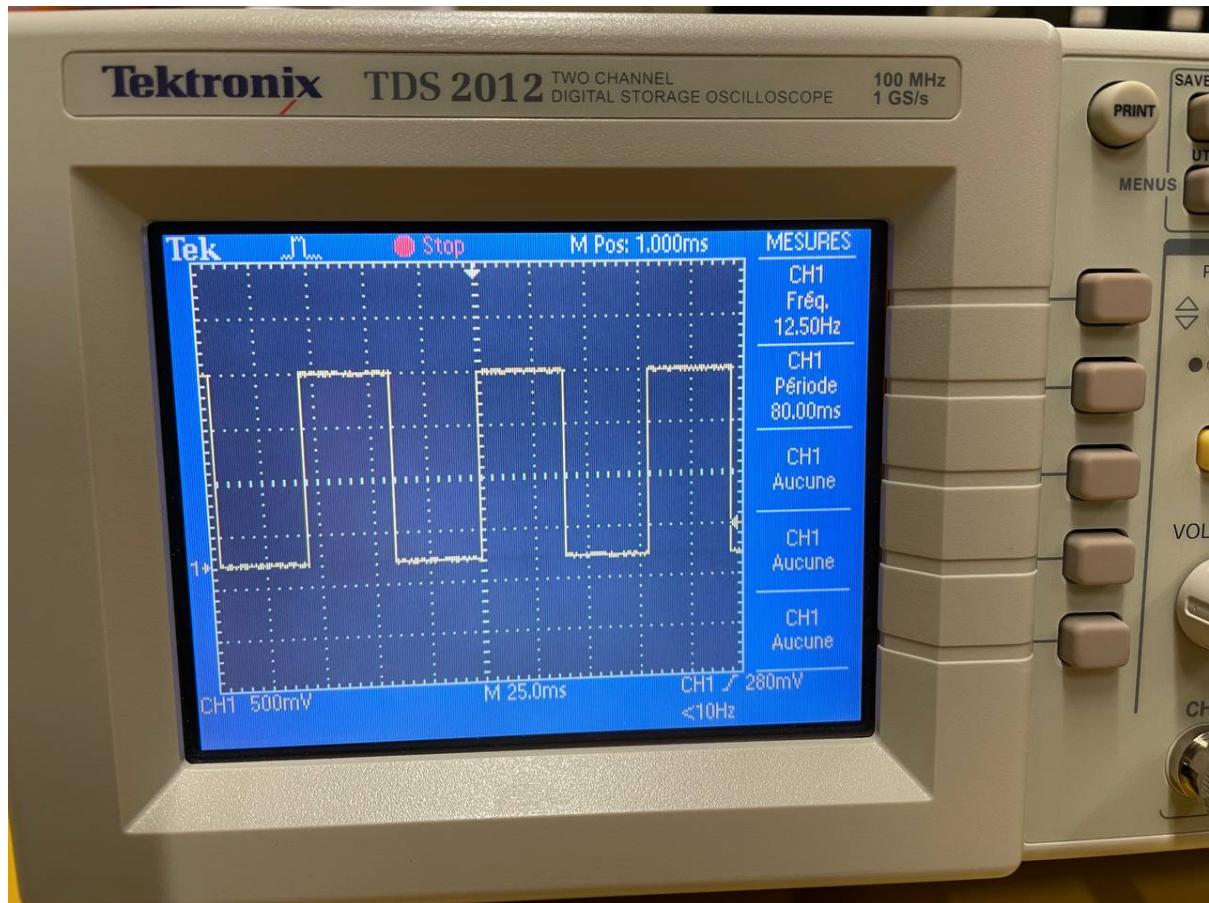
Canal de lecture

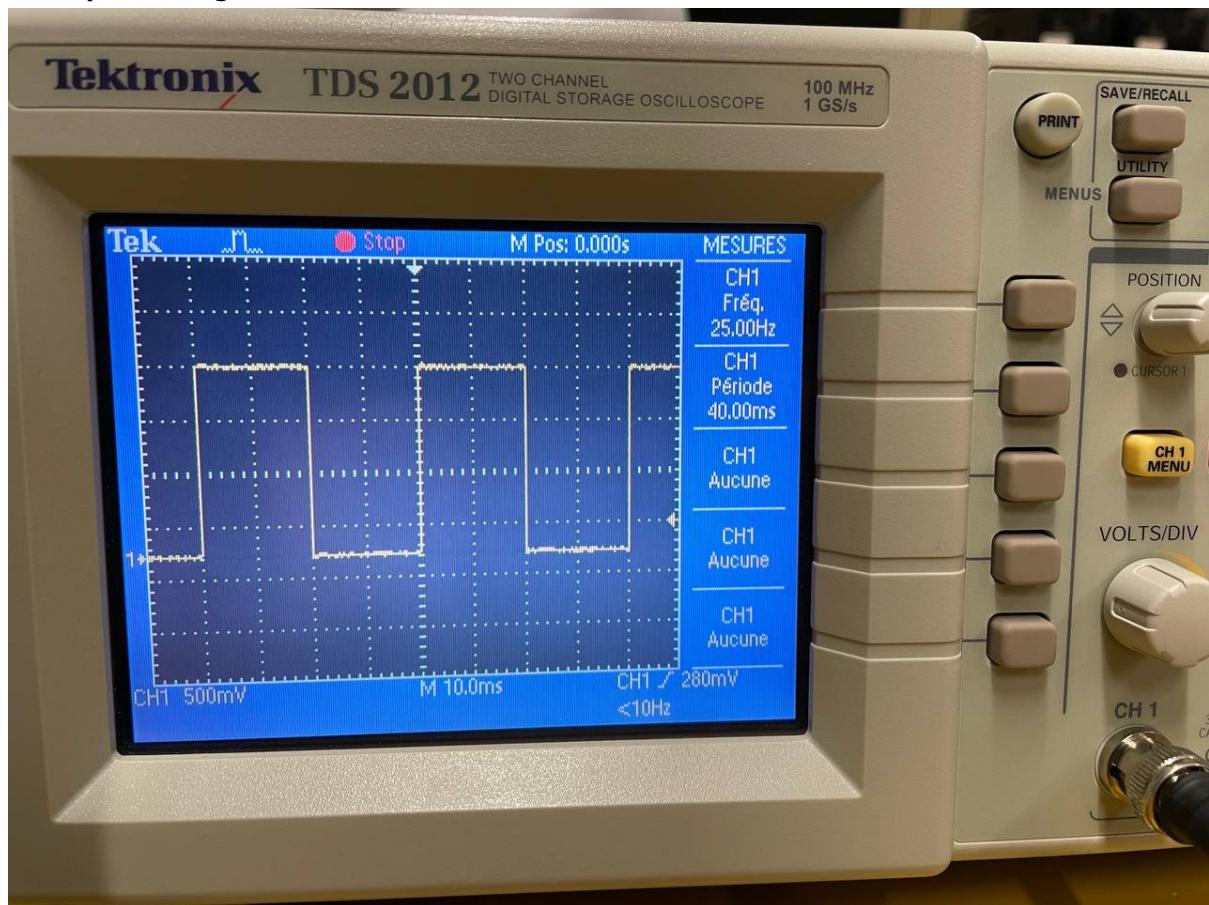
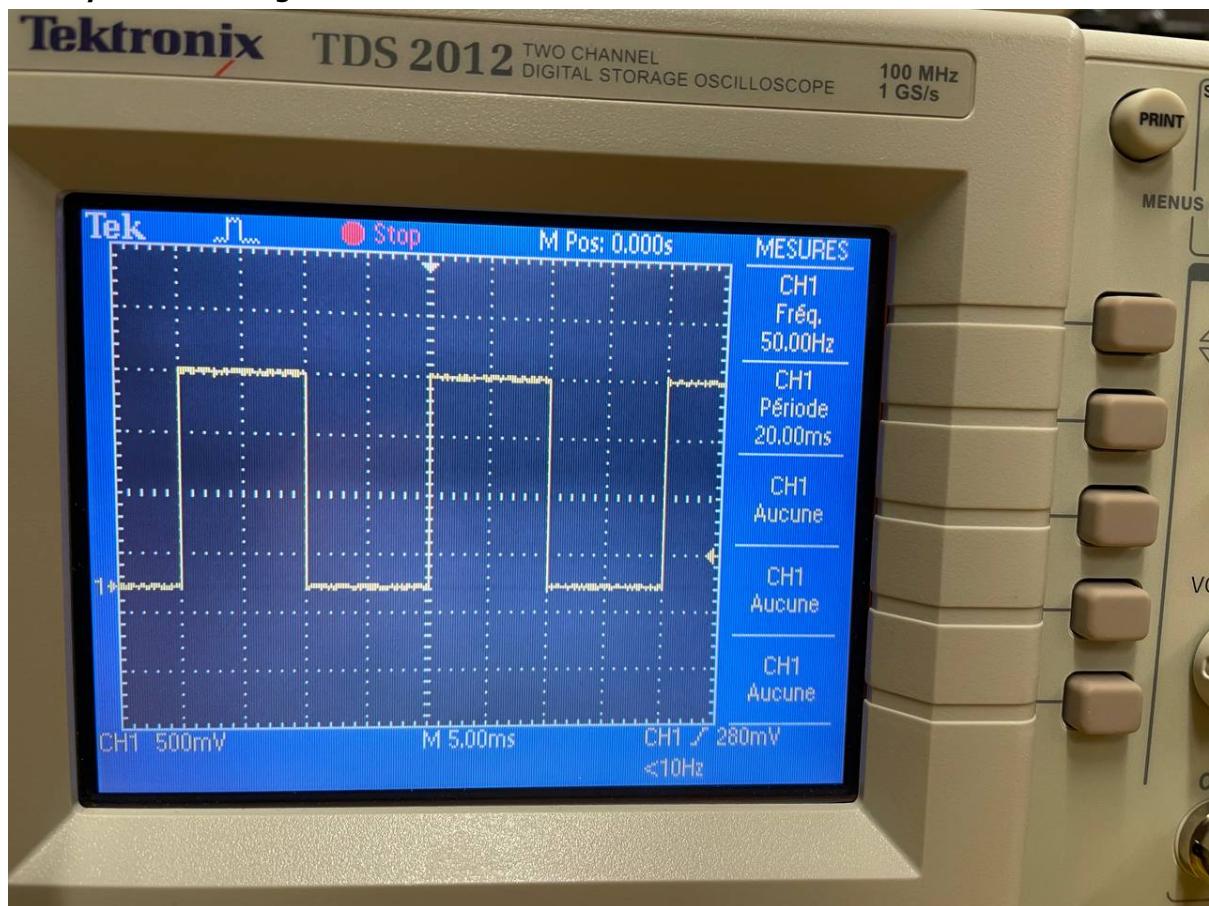


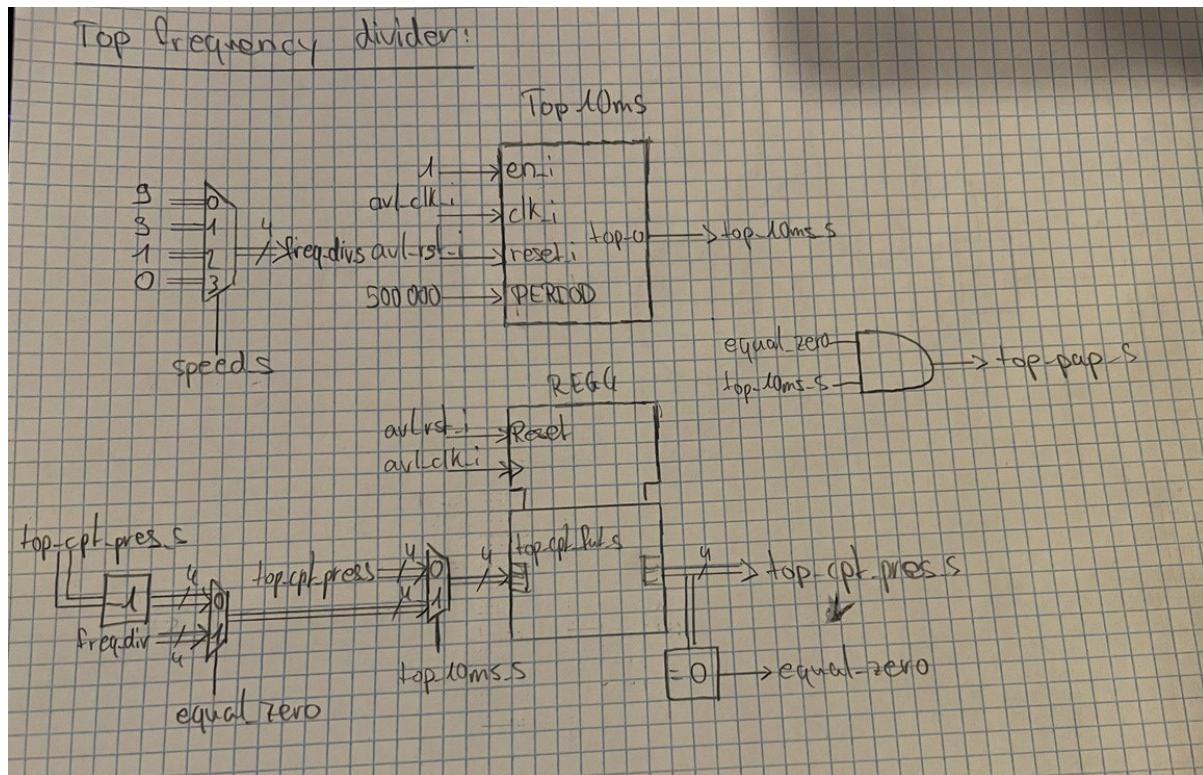
Générateur de top



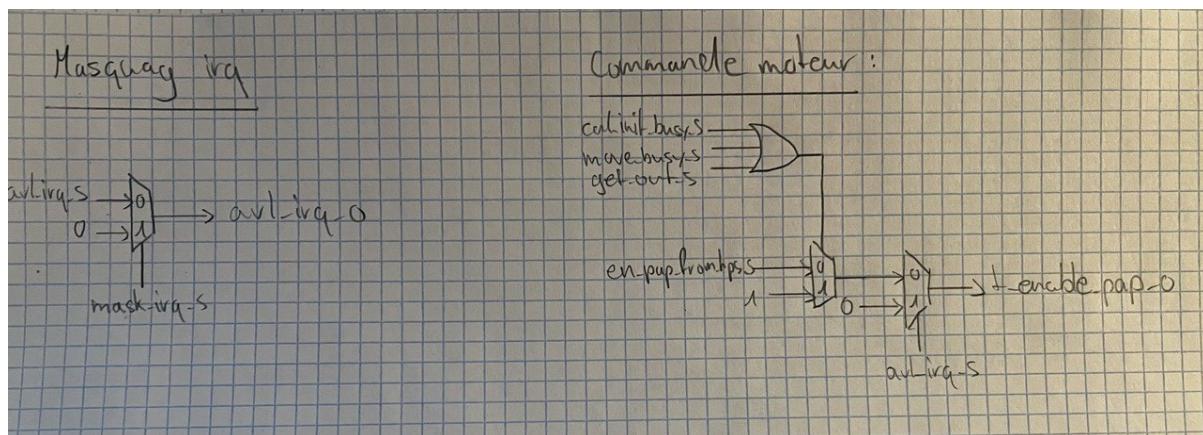
Voici les mesures à l'oscilloscope pour les différentes vitesses de la table tournante (les périodes sont doublées) :

5 Hz / Viteese petite**12.5 Hz / Vitesse moyenne**

25 Hz / Vitesse grande**50 Hz / Vitesse très grande****Diviseur de fréquence**

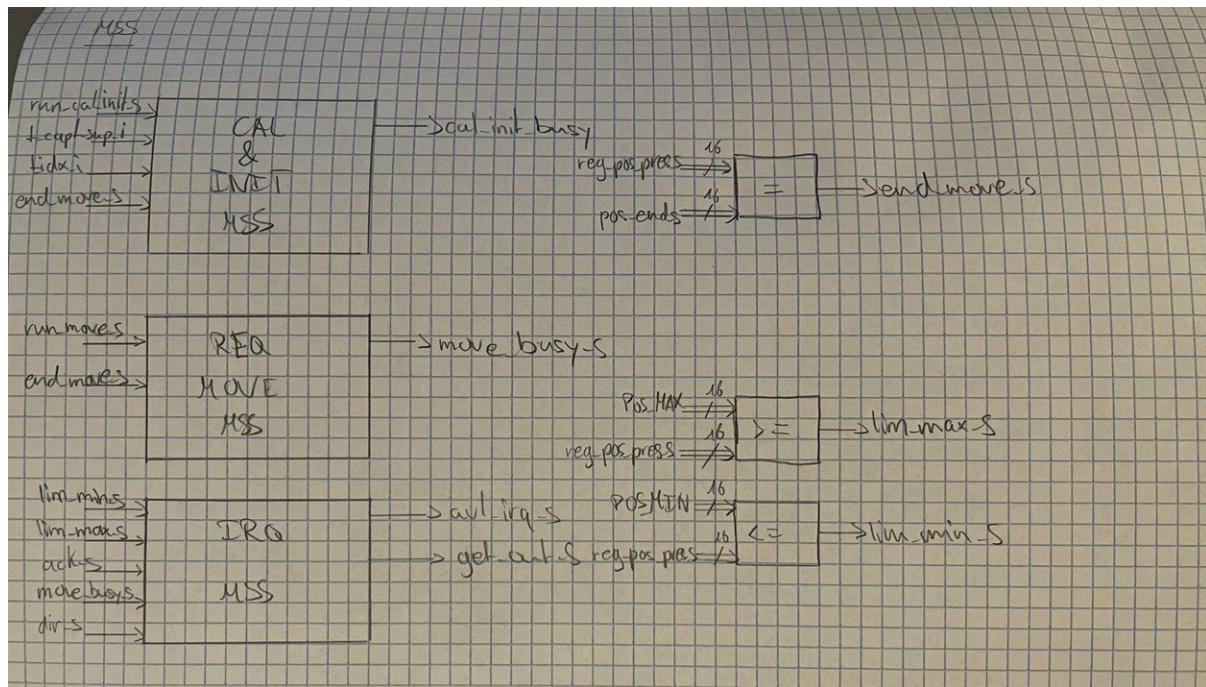


Masquage des IRQ et commande moteur

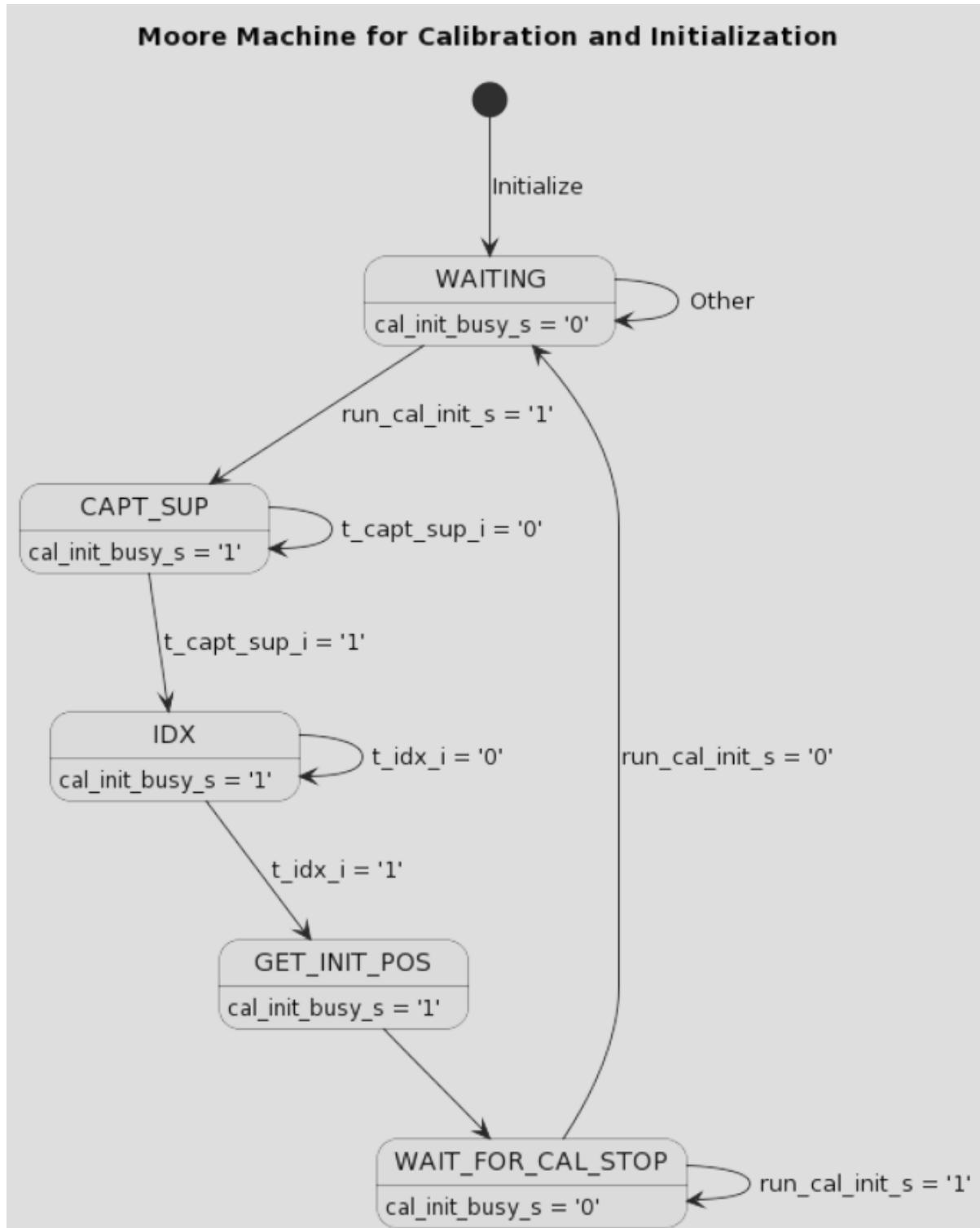


Machine d'état

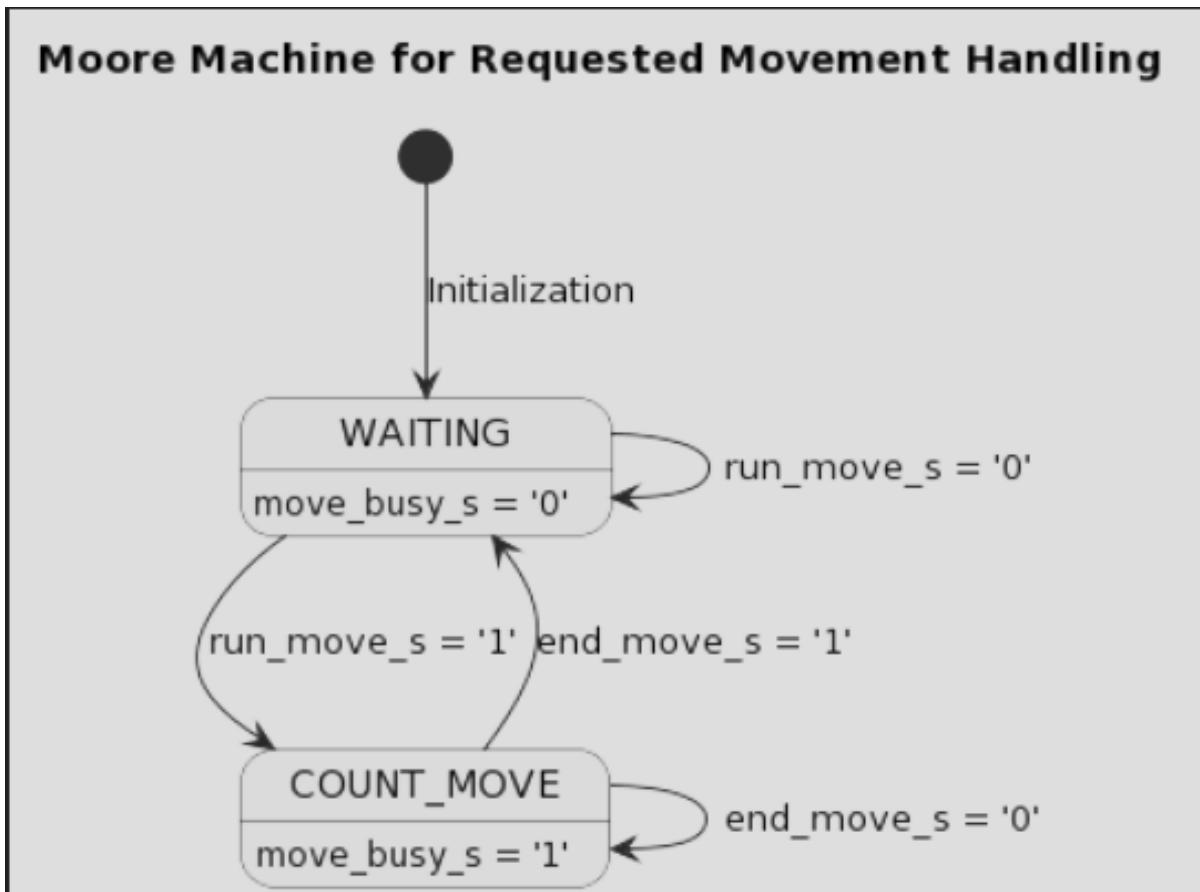
La solution proposée pour le laboratoire utilise 3 machines d'état.



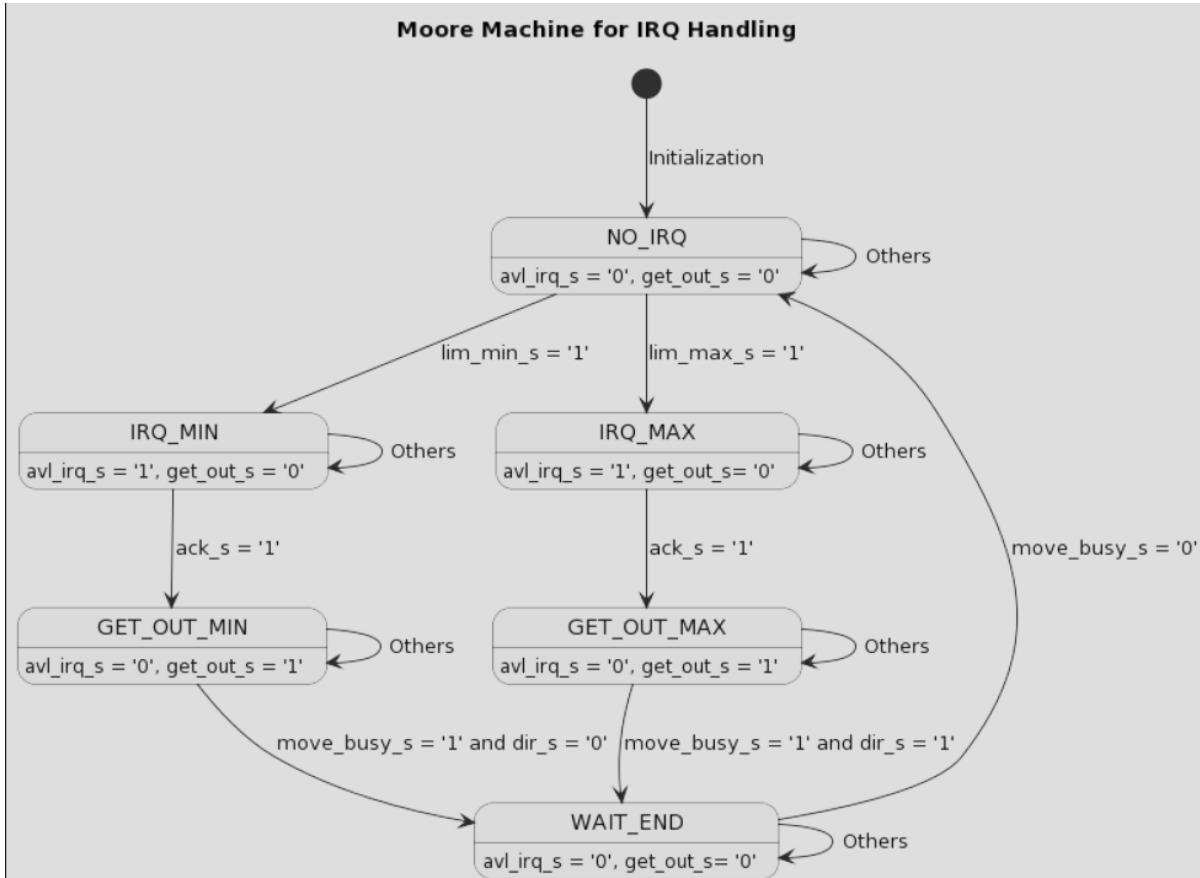
Machine d'état pour la calibration/initialisation



Machine d'état pour le déplacement



Machine d'état pour les IRQ



UART

L'UART requiert certains paramètres pour fonctionner correctement. Voici les paramètres utilisés pour ce laboratoire :

1. Baudrate: Le baudrate doit être défini à 9600. Pour calculer les valeurs de Divisor Latch Low (DLL) et Divisor Latch High (DLH), vous pouvez utiliser la formule fournie dans la documentation :

$$\text{baud rate} = \frac{\text{serial clock frequency}}{16 \times \text{divisor}}$$

Puisque l'horloge l4_sp_clk est à 100 MHz, le diviseur à utiliser pour obtenir un baudrate de 9600 est :

$$\text{divisor} = \frac{100,000,000}{16 \times 9600} \approx 651$$

2. Bit de données: Pour définir le nombre de bits de données à 8, vous devrez vous assurer que le bit DLAB (Divisor Latch Access Bit) du registre LCR (Line Control Register) est réglé sur 0 pour désactiver l'accès aux registres du diviseur. Ensuite, configurez les bits de données du registre LCR pour 8 bits.
3. Bit de parité désactivé: Dans le registre LCR, assurez-vous que les bits de parité sont désactivés, ce qui est généralement le réglage par défaut.
4. Bit de stop: Configurez le registre LCR pour un seul bit de stop. Activer les buffers FIFO en émission et réception: Activez les FIFO en écrivant dans le registre FCR (FIFO Control Register).

Synthèse

Analysis & Synthesis Status	Successful - Sun Jan 28 09:43:06 2024
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Standard Edition
Revision Name	DE1_SoC_top
Top-level Entity Name	DE1_SoC_top
Family	Cyclone V
Logic utilization (in ALMs)	N/A
Total registers	825
Total pins	317
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	1

La synthèse montre que le système tourne avec 825 registres ce qui est beaucoup comparé aux laboratoires précédents. Le professeur a cependant indiqué lors du laboratoire précédent que le nombre de registres compté par quartus n'était pas fiable à 100%.

Conclusion

En conclusion de ce projet de conception d'une table tournante avec une interface avl_user_interface, les objectifs ont été atteints. La mise en œuvre de l'interface UART0 du HPS a été réalisée avec succès, en s'appuyant sur la documentation du Cyclone V.

Le plan d'adressage de l'interface avl_user_interface a été modifié progressivement tout le long du laboratoire. La partie initialisation/calibration ont été les seules fonctionnalités complexes décrites en VHDL plutôt que codées en C. Malgré la préférence d'intégrer le code complexe en soft plutôt qu'en hard, la latence du bus ne permet pas d'atteindre la précision souhaitée si codée côté soft.

Le générateur de top a été intégré avec succès dans l'interface, et la fréquence du signal top a été vérifiée à l'oscilloscope pour respecter la contrainte de ne jamais dépasser 100 Hz.

L'unité de commande de la table tournante a été définie et conçue avec une interface semblable aux laboratoires précédents.

Les interruptions, gérées par un gestionnaire d'interruption conforme aux spécifications, ont été implémentées avec succès. La routine d'interruption fpga_ISR() a été testée avec le toggle de la Led7 de la DE1-SoC.