



CONCEPTION DE SYSTÈME NUMÉRIQUE  
CSN

---

JOURNAL DU COMMANDE DE DÉPLACEMENT D'UN  
CHARIOT  
LABORATOIRE MSS COMPLEXE

---

Domingues Pedrosa João Miguel  
Nicolas Kobel

Professeur: M. Messerli Etienne

24 janvier 2016

## Table des matières

<b>1 Mandat</b>	<b>2</b>
<b>2 Analyse</b>	<b>3</b>
2.1 Commande des moteurs . . . . .	3
2.1.1 Organigramme . . . . .	3
2.1.2 MSS génération de commande . . . . .	4
<b>3 Schéma</b>	<b>5</b>
<b>4 Simulation</b>	<b>6</b>
<b>5 Synthèse</b>	<b>8</b>
<b>6 VHDL</b>	<b>9</b>

# 1 Mandat

Le but de ce laboratoire est de commander la rotation d'un moteur. Il y a deux modes. Le mode manuel qui fait tourner le moteur tant que l'on reste appuyer sur le bouton start et le mode automatique où l'on avance du nombre de position définit. Il sera possible de modifier la vitesse de rotation

Schéma des entrées :

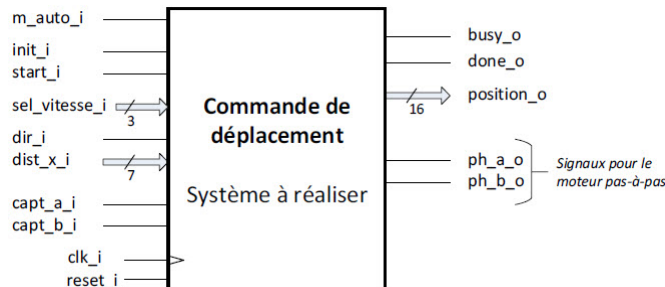


Tableau des entrées/sorties :

Port	Direction	Taille	Description
<b>m_auto_i</b>	entrée	1	Mode de fonctionnement du système. <ul style="list-style-type: none"> <li>• '0' mode manuel</li> <li>• '1' mode automatique</li> </ul>
<b>init_i</b>	entrée	1	Signal d'initialisation de la position (position = 0)
<b>start_i</b>	entrée	1	Signal d'activation du déplacement du chariot
<b>sel_vitesse_i</b>	entrée	3	Signal de sélection de la vitesse de déplacement du chariot. Voir tableau de correspondance.
<b>dir_i</b>	entrée	1	Signal indiquant la direction du déplacement <ul style="list-style-type: none"> <li>• '0' déplacement positif</li> <li>• '1' déplacement négatif</li> </ul>
<b>dist_x_i</b>	entrée	7	Entrée indiquant la distance du déplacement à réaliser en mode automatique. Cette valeur sera multipliée par 8 pour le déplacement de la table
<b>capt_a_i</b>	entrée	1	Signal venant du codeur incrémental permettant de mesurer la position du chariot (table tournante) <ul style="list-style-type: none"> <li>• signal A du codeur</li> </ul>
<b>capt_b_i</b>	entrée	1	Signal venant du codeur incrémental permettant de mesurer la position du chariot (table tournante) <ul style="list-style-type: none"> <li>• signal B du codeur</li> </ul>
<b>clock_i</b>	entrée	1	Horloge du système synchrone, 1 MHz
<b>reset_i</b>	entrée	1	Signal de remise à zéro asynchrone
<b>busy_o</b>	sortie		Signal indiquant qu'un déplacement est en cours
<b>done_o</b>	sortie	1	Signal indiquant que le déplacement demandé est terminé
<b>position_o</b>	sortie	16	Sortie de 16 bits indiquant la position du chariot (table tournante). La valeur est un entier positif.
<b>ph_a_o</b>	sortie	1	Signal de commande du moteur pas-à-pas. Une séquence spécifique sur ph_a et ph_b permet de faire tourner le moteur dans le sens horaire ou antihoraire <ul style="list-style-type: none"> <li>• phase-A du moteur</li> </ul>
<b>ph_b_o</b>	sortie	1	Signal de commande du moteur pas-à-pas. Une séquence spécifique sur ph_a et ph_b permet de faire tourner le moteur dans le sens horaire ou antihoraire <ul style="list-style-type: none"> <li>• phase-B du moteur</li> </ul>

Tableau des vitesses :

sel_vitesse_i	pas par sec	période ms
"000"	1.00	1000
"001"	2.00	500
"010"	5.00	200
"011"	10.00	100
"100"	16.67	60
"101"	25.00	40
"110"	50.00	20
"111"	100.00	10

Chronogramme pour commander le moteur :

Signaux de commande du moteur pas-à-pas



## 2 Analyse

### 2.1 Commande des moteurs

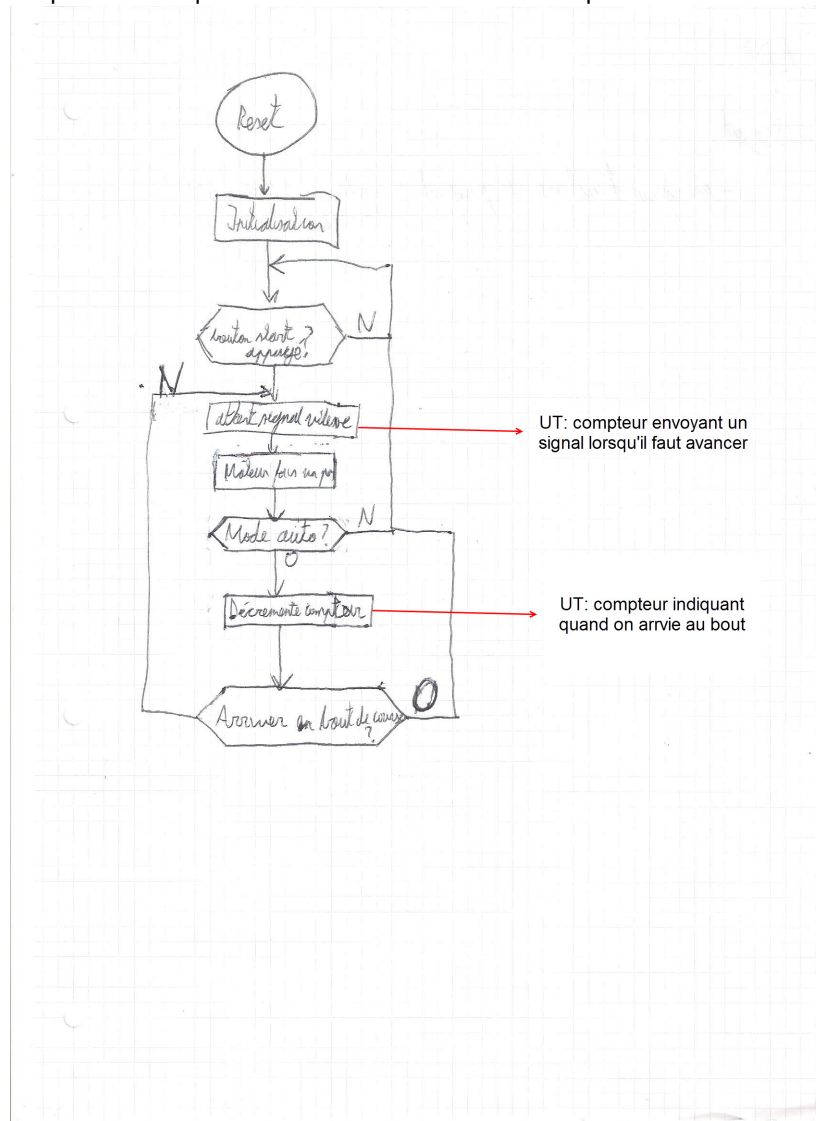
Pour la commande des moteurs, nous avons fait la séparation suivante. La vitesse correspond à un diviseur d'horloge qui dépendra de la valeur de celui-ci. Pour le parcours de distance en mode automatique, il s'agit que d'un compteur dont on charge la valeur définit au début du cycle et qui va décrémenter à chaque pas. Les pas sont dépendant de la vitesse, il faudra donc utilisé la sortie du diviseur d'horloge pour avoir la référence. Il génère une pulse lorsque l'on arrive à la fin de la distance.

La génération de commande se fait via une MSS. Elle reçoit en entrée le signal `start`, la direction, le mode (manuel/auto), la fin de distance pour le mode automatique et le diviseur d'horloge pour la vitesse. Il indique en sortie si le moteur est occupé ou non et fait les différent changement de phase pour le moteur. Il s'occupe donc de faire les cycle moteur comme indiqué sur le chronnogramme lorsque celui si est en marche.

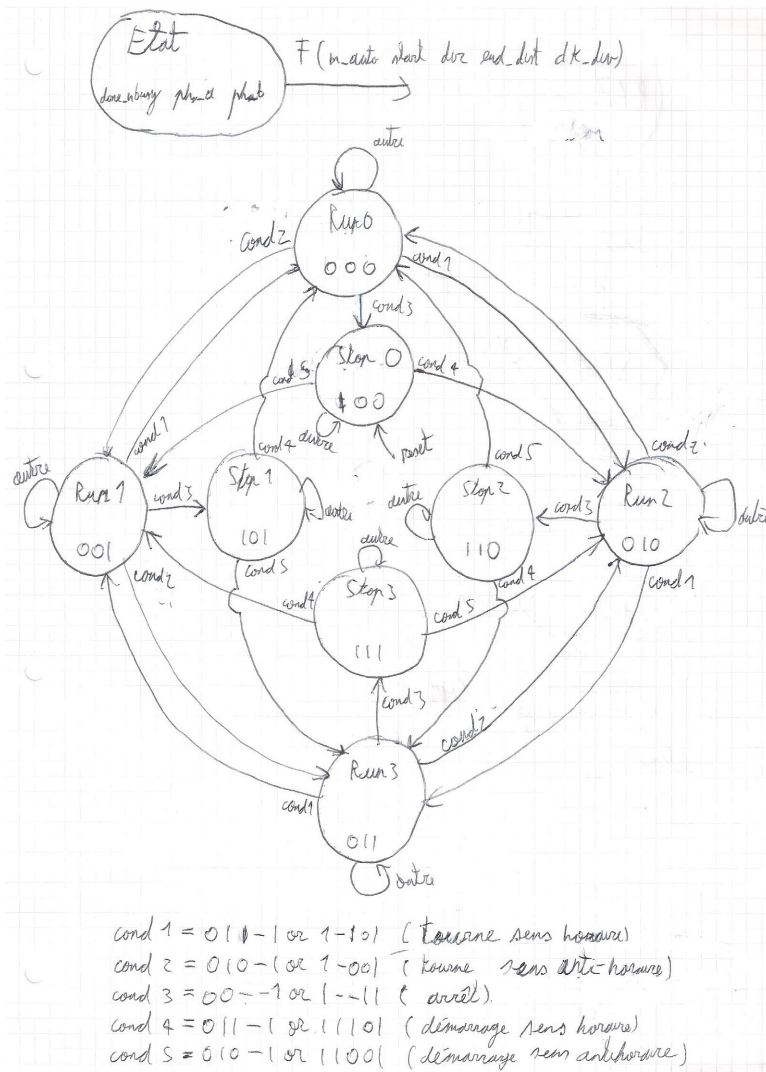
L'acquisition de position a été repris de l'ancien laboratoire de Domingues Pedrosa João Miguel.

#### 2.1.1 Organigramme

Il faut faire attention au lorsque l'on avance d'un pas, il faut faire attention d'attendre sur le diviseur de fréquence afin que l'on est bien la bonne vitesse pour la rotation du moteurs.



## 2.1.2 MSS génération de commande



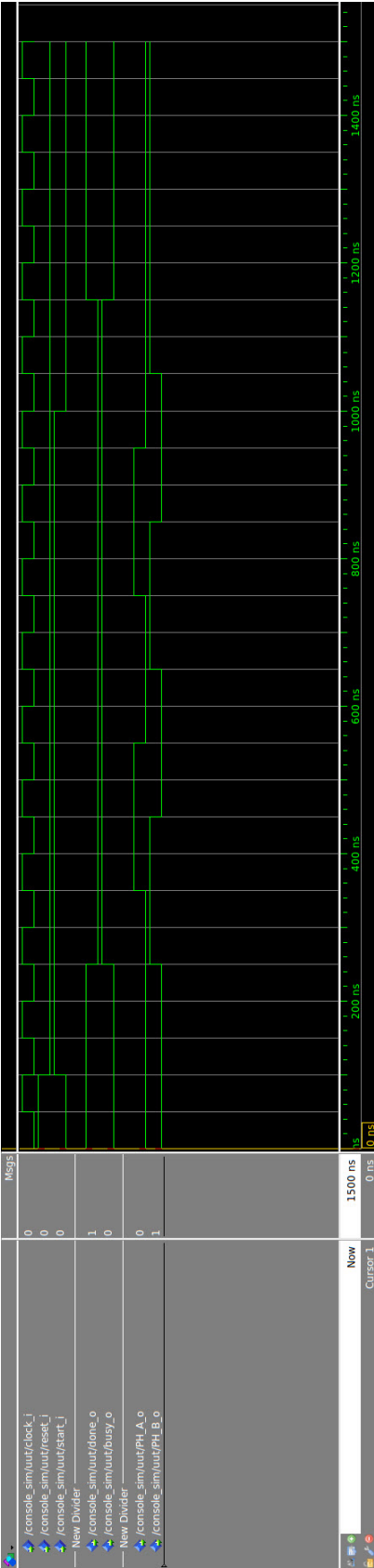
### **3 Schéma**

## 4 Simulation

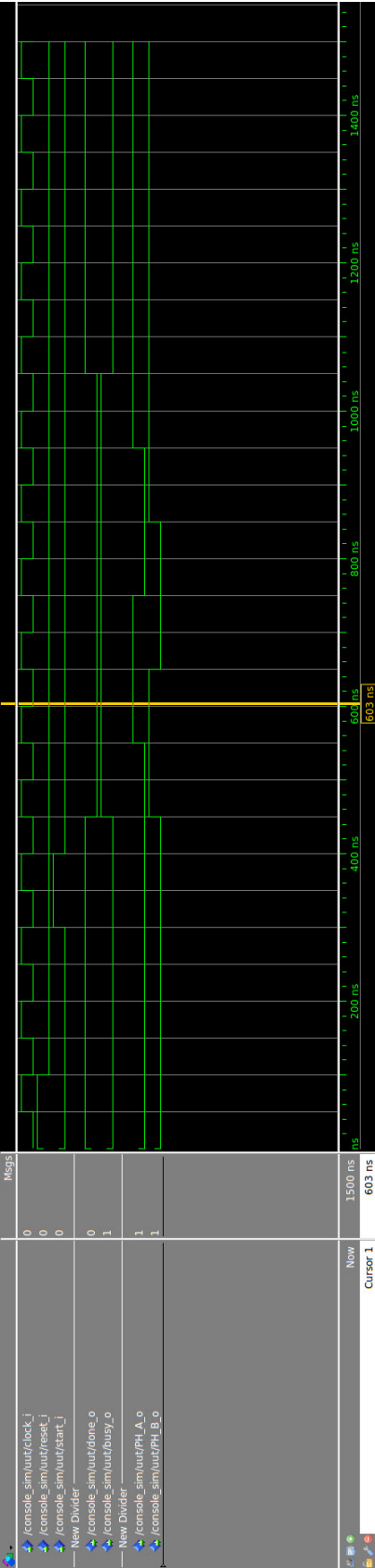
Pour les tests, nous avons testé que le fonctionnement normal dans les deux modes. En mode manuel, nous avons fait forcer le `div_clk` à un afin de voir lorsque l'on appuie sur `start` les sortie `ph_a` et `ph_b` changeait et exécutait la bonne séquence. Le mode automatique, nous avons vérifier qu'après un flanc du `start` les sorties changer le bon nombre de fois. Durant c'est deux tests, nous avons aussi vérifier que les sorties `busy` et `done` correspondant au état attendu durant que le moteur fonctionne ou est à l'arrêt. Par manque de temps des tests plus poussé n'ont pas pu être effectuer. Par exemple, nous n'avons pas pus tester l'initialisation avec l'entrée `init` ou les deux sens de rotation, ici nous avons tester quand sens anti-horaire.

Durant les tests, nous nous somme aperçu de deux erreurs d'implémentation. La première est que lorsque l'on est en mode automatique, il faut laisser le bouton `start` appuyer jusqu'au prochain signal du `clk_div`. Cela peut durer jusqu'à une seconde selon la vitesse sélectionner. La deuxième est que l'on fait un pas de plus que le nombre définit en mode automatique. Par manque de temps, nous n'avons pas pu les corriger.

Chronogramme (Mode Manuel) :



Chronogramme (Mode Auto) :





## 5 Synthèse

Compilation Report - cmd\_mvnt\_mot\_top

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global S
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
  - Summary
  - Settings
    - Parallel Compilation
    - Source Files Read
    - Resource Usage Summ
    - Resource Utilization by
  - State Machines
  - Optimization Results
  - Parameter Settings by
  - Connectivity Checks
  - Messages
- Fitter
  - Summary
  - Settings
    - Parallel Compilation
    - Pin-Out File
  - Resource Section
    - Resource Usage Sur
    - Input Pins
    - Output Pins
    - I/O Bank Usage
    - All Package Pins
    - Output Pin Default L
    - Resource Utilization
    - Delay Chain Summa
    - Control Signals
    - Global & Other Fast
    - Non-Global High Far

Fitter Resource Usage Summary

	Resource	Usage
1	Total logic elements	129 / 570 ( 23 % )
1	-- Combinational with no register	46
2	-- Register only	7
3	-- Combinational with a register	76
2		
3	Logic element usage by number of LUT inputs	
1	-- 4 input functions	47
2	-- 3 input functions	35
3	-- 2 input functions	34
4	-- 1 input functions	5
5	-- 0 input functions	1
4		
5	Logic elements by mode	
1	-- normal mode	90
2	-- arithmetic mode	39
3	-- qfbk mode	9
4	-- register cascade mode	0
5	-- synchronous clear/load mode	50
6	-- asynchronous clear/load mode	83
6		
7	Total registers	83 / 570 ( 15 % )
8	Total LABs	21 / 57 ( 37 % )
9	Logic elements in carry chains	42
10	Virtual pins	0
11	I/O pins	38 / 159 ( 24 % )
1	-- Clock pins	2 / 4 ( 50 % )
12		
13	Global signals	3
14	UFM blocks	0 / 1 ( 0 % )
15	Global clocks	3 / 4 ( 75 % )
16	JTAGs	0 / 1 ( 0 % )
17	Average interconnect usage (total/H/V)	6% / 8% / 4%
18	Peak interconnect usage (total/H/V)	6% / 8% / 4%
19	Maximum fan-out	83
20	Highest non-global fan-out	25

for setup requirements

Lors de l'importation sur la carte, nous avons tester le fonctionnement manuel et automatique. Nous avons aussi essayer les deux direction et différentes vitesses.

## **6 VHDL**