



PROJET FGPA M1

RAPPORT

Centrale DCC sur FPGA



Auteurs

BISLIEV MAGOMED
YANG ZIXIAO

Encadrant

RHOUNI AMINE

2022-2023

Note : Vous trouverez une vidéo de démonstration à la fin avec le Bilan d'avancement

Sommaire

I	Introduction	2
II	Architecture de la centrale DCC	3
II.1	Générateurs DCC Bit 1 et 0	3
II.1.1	Description du module	3
II.1.2	Simulation du module	4
II.2	Registre DCC	4
II.2.1	Description du module	4
II.2.2	Simulation du module	4
II.3	MAE Générale	5
II.3.1	Description du module	5
II.3.2	Simulation du module	6
II.4	Generateur de trames de test	7
II.4.1	Module de validation du reste de l'architecture DCC	7
II.4.2	Remplacement par le Microblaze	7
II.5	Afficheur sept segment	9
II.5.1	Description du module	9
III	Ajout de la centrale DCC comme IP dans un système Microblaze	10
III.1	Organigramme du code C du Microblaze	10
III.2	Guide d'utilisation	11
III.2.1	Interrupteurs	11
III.2.2	Boutons	11
III.2.3	Sélection de l'adresse du train	11
III.2.4	Activation de la marche avant	11
III.2.5	Activation de la marche arrière	12
III.2.6	Activation d'une fonction	12
III.2.7	Désactivation d'une fonction	12
III.2.8	Reset de la mémoire du train	12
IV	Bilan d'avancement et démonstration	13

I Introduction

Dans le cadre de ce projet, nous avons réalisé une centrale de commande afin de contrôler un train. La commande du train se fait en utilisant le protocole DCC qui consiste à transmettre une commande sous forme d'une trame numérique où chaque élément binaire doit être représenté par une impulsion à 0 suivie d'une impulsion à 1 avec des durées différentes selon qu'il s'agit d'un 1 ou un 0. L'ordre des impulsions doit être respecté et une trame doit être envoyée toutes les 6ms. Chaque trame contient un preambule, un bit start entre chaque champ, un champ d'adresse, un champ de commande, un champ de contrôle et un bit stop. Le champ de commande contrôle la vitesse du train et active ou désactive une fonction par exemple les phares ou des effets sonores, le champ de contrôle est un xor entre l'adresse et la commande afin de vérifier que la trame est valide.

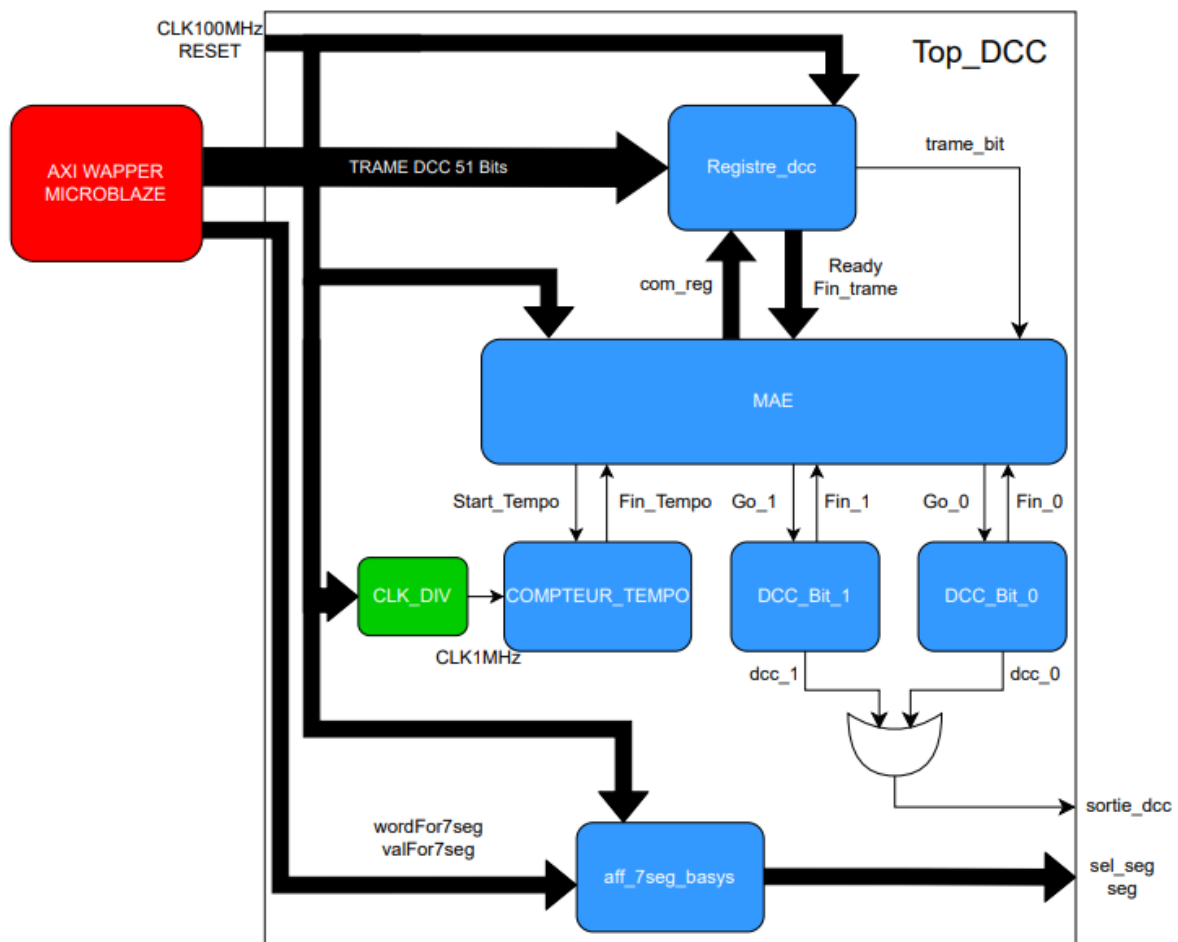


Fig. 1. Centrale DCC

II Architecture de la centrale DCC

II.1 Générateurs DCC Bit 1 et 0

II.1.1 Description du module

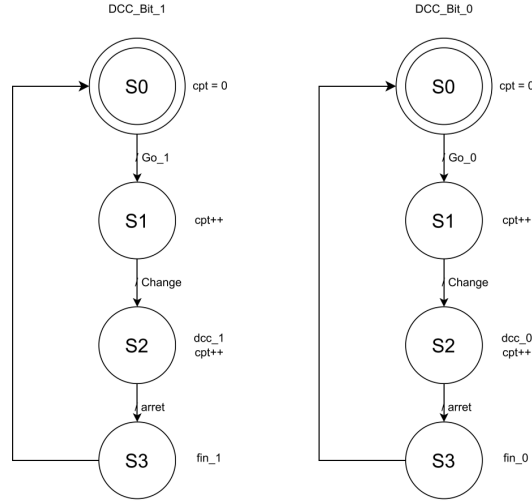


Fig. 2. Machine à états de DCC Bit 1 (à gauche) et 0 (à droite)

Etat	Transition	Action
S0		État de départ : on initialise un compteur à 0 La sortie dcc est à 0
	Go_1 (Bit1) Go_0 (Bit0)	Début de génération du signal Passage à l'état S1
S1		Incrémentation du compteur à chaque front montant d'horloge
	Change	Le compteur atteint (58:Bit1, 100:Bit0) Passage à l'état S2
S2		La sortie dcc passe à 1 Incrémentation du compteur à chaque front montant d'horloge
	Arret	Le compteur atteint (116:Bit1, 200:Bit0) Passage à l'état S3
S3		Fin de génération de bit, Fin_1(Bit1) ou Fin_0(Bit0) passent à 1

Tableau 1: Transitions et états

II.1.2 Simulation du module

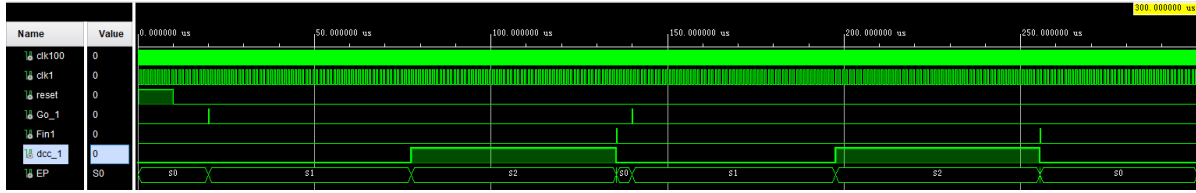


Fig. 3. Simulation de la machine à états de DCC Bit 1 et 0

L'entrée Go_1 (Go_0 : Bit 0) marque le début du Bit en générant une impulsion à 0 pendant $58\mu s$ ($100\mu s$: Bit 0) puis une impulsion à 1 pendant $58\mu s$ ($100\mu s$: Bit 0), l'entrée Fin_1 (Fin_0:Bit 0) marque la fin de la génération du Bit .

II.2 Registre DCC

II.2.1 Description du module

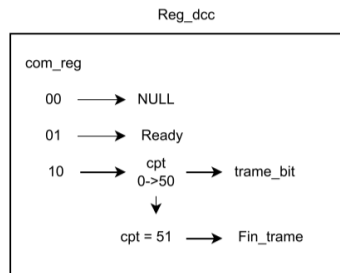


Fig. 4. Analyse Registre DCC

II.2.2 Simulation du module

com_reg	Action
01	Chargement de la trame Mise à 1 de Ready
10	Envoi de la trame bit par bit (Décalage) Mise à 1 de Fin_trame si le compteur atteint 51

Tableau 2. Description du module en fonction de l'état de la commande

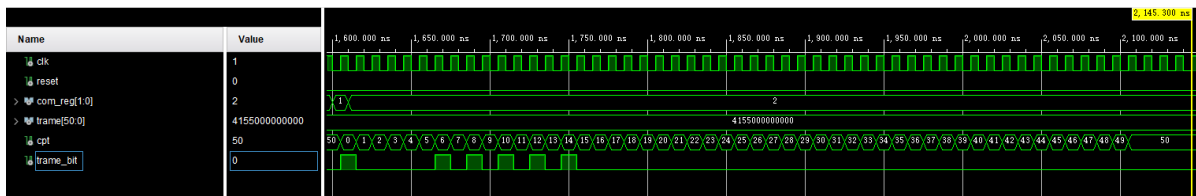


Fig. 5. Simulation de Registre DCC

Au départ la commande vaut **01**, ce qui entraîne un chargement de la trame sur 51 bits qui se trouve en entrée, ici la trame est **41550000000000** en base hexadécimale, soit en binaire **10000010101010100....**. Ensuite, la commande passe à **10**, cela provoque le déclenchement du décalage progressif de la trame en allant du bit de poids fort au bit de poids faible. Durant un cycle d'horloge on transmet un **1**, les 5 cycles suivants on transmet un **0**, puis on retransmet un **1**, ainsi de suite pour le reste de la trame.

II.3 MAE Générale

II.3.1 Description du module

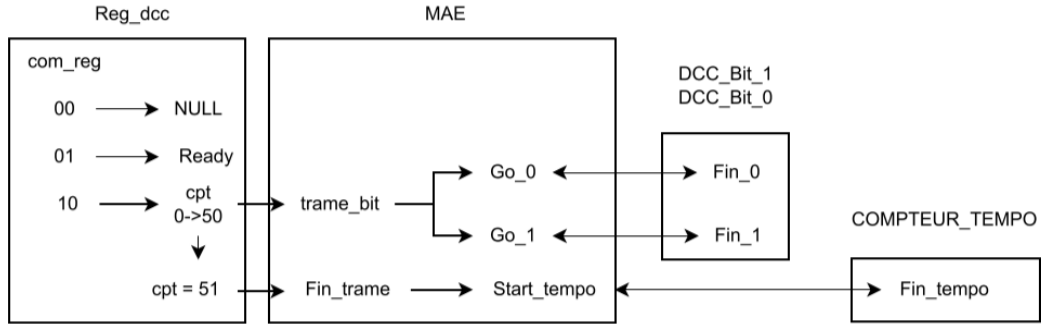


Fig. 6. Interaction de la MAE avec les autres modules

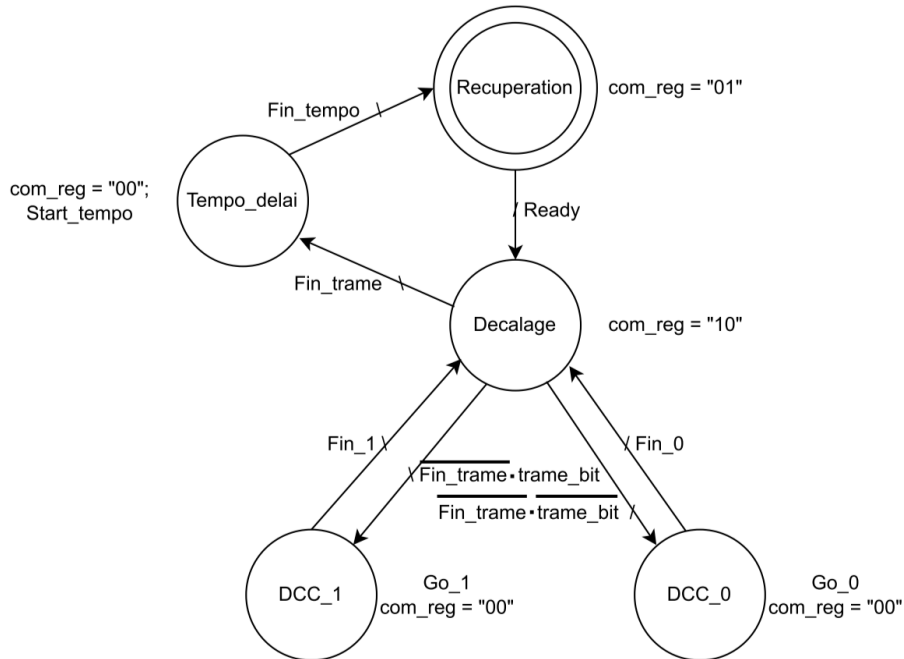


Fig. 7. Machine à états Générale

Etat	Transition	Action
Récupération		État de départ : récupération de la trame en entrée du module Registre_dcc en positionnant com_reg à 01
	Ready	La trame a été récupérée par le module Registre_dcc Passage à l'état Décalage
Décalage		La MAE récupère bit à bit la trame
	trame_bit='0' and Fin_trame='0'	Le Bit est un 0 et on n'est pas à la fin de la trame Passage à l'état DCC_0
	trame_bit='1' and Fin_trame='0'	Le Bit est un 1 et on n'est pas à la fin de la trame Passage à l'état DCC_1
	Fin_trame='1'	On est à la fin de la trame Passage à l'état Tempo_delai
DCC_0		Génération d'un Bit 0 On ne récupère pas de nouveau Bit pendant la génération d'un Bit 0 (com_reg=00)
	Fin_0	La génération d'un Bit 0 est terminée Passage à l'état Décalage
DCC_1		Génération d'un Bit 1 On ne récupère pas de nouveau Bit pendant la génération d'un Bit 1 (com_reg=00)
	Fin_1	La génération d'un Bit 1 est terminée Passage à l'état Décalage
Tempo_delai		Début de comptage de 6ms (Start_Tempo='1')
	Fin_tempo='1'	Au bout de 6ms Fin_Tempo passe à 1 Passage à l'état Récupération, pour récupérer une nouvelle trame

Tableau 3: Transitions et états

II.3.2 Simulation du module

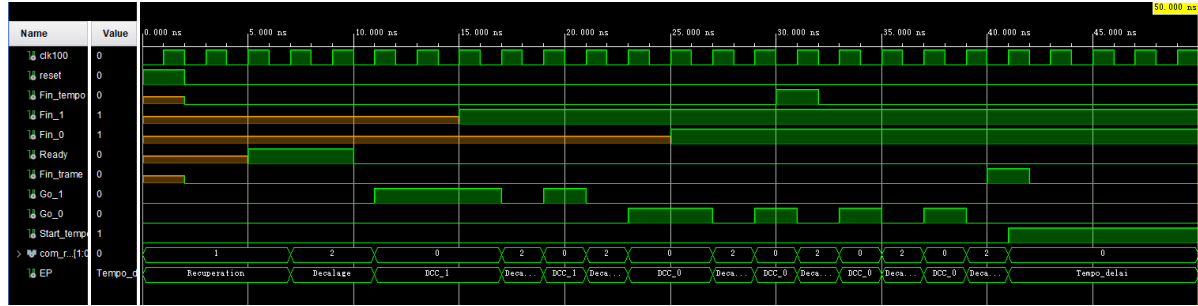


Fig. 8. Simulation de la Machine à états générale

Au départ nous sommes dans l'état **Récupération** c'est-à-dire en attente du passage à 1 de *Ready*, qui indique qu'une trame a pu être récupérée, la détection de *Ready* entraîne le passage dans l'état **Décalage**, *Go_1* indique que le bit à l'entrée est un 1, cela provoque un passage dans l'état **DCC_1**. *Fin_1* indique la fin de la génération de Bit 1, on retourne dans l'état **Décalage**. En fonction de si la donnée en entrée est un 1 ou un 0 on retrouve le même enchaînement d'états et ce jusqu'à qu'on arrive à la fin de la trame qui est indiqué par *Fin_trame*, cela provoque le passage à l'état **Tempo_delai** pour déclencher le comptage de 6ms en mettant *Start_Tempo* à 1, lorsque le compteur atteint 6ms, on retourne dans l'état **Récupération**.

II.4 Générateur de trames de test

Dans un premier temps le fonctionnement de l'architecture est vérifié à l'aide d'un module "DCC Frame Generator" qui en fonction de l'état de 8 Interrupteurs génère différentes trames:

Interrupteur actif	Commande
7	Activation de la marche avant du train d'adresse 1
6	Activation de la marche arrière du train d'adresse 1
5	Allumage des Phares du Train d'adresse 1
4	Extinction des Phares du Train d'adresse 1
3	Activation du Klaxon (Fonction F11) du Train d'Adresse 1
2	Réamorçage du Klaxon (Fonction F11) du Train d'Adresse 1
1	Activation de l'annonce SNCF (Fonction F13) du Train d'Adresse 1
0	Extinction de l'annonce SNCF (Fonction F13) du Train d'Adresse 1
Aucun	Arrêt du Train d'Adresse 1

Tableau 4: Commandes envoyées en fonction de l'état des interrupteurs

II.4.1 Module de validation du reste de l'architecture DCC

Simulation

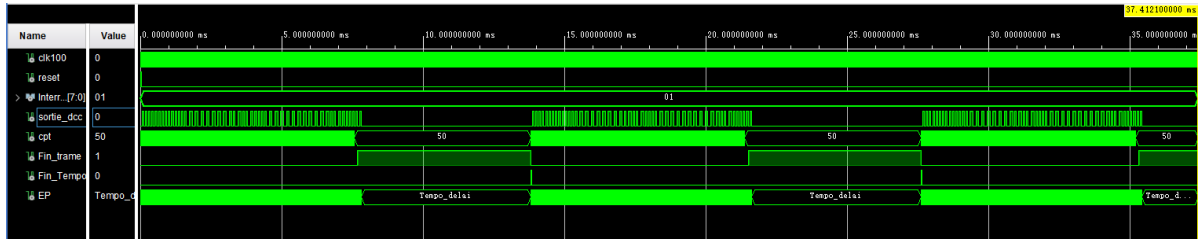


Fig. 9. Simulation de Top DCC

Les trames respectent bien la structure imposée par le cahier des charges et sont séparées d'une durée de 6ms.

II.4.2 Remplacement par le Microblaze

La Centrale DCC est par la suite intégrée dans un système Microblaze en la transformant en IP pour interagir avec le bus AXI. L'architecture du système comprend la gestion des entrées/sorties de la carte Basys3 par le Microblaze à l'aide de contrôleurs GPIO. Dans cette étape le module Frame Generator est retiré et la trame sera récupérée via les registres de l'AXI Wrapper comme indiqué sur le schéma de la **Figure 1**.

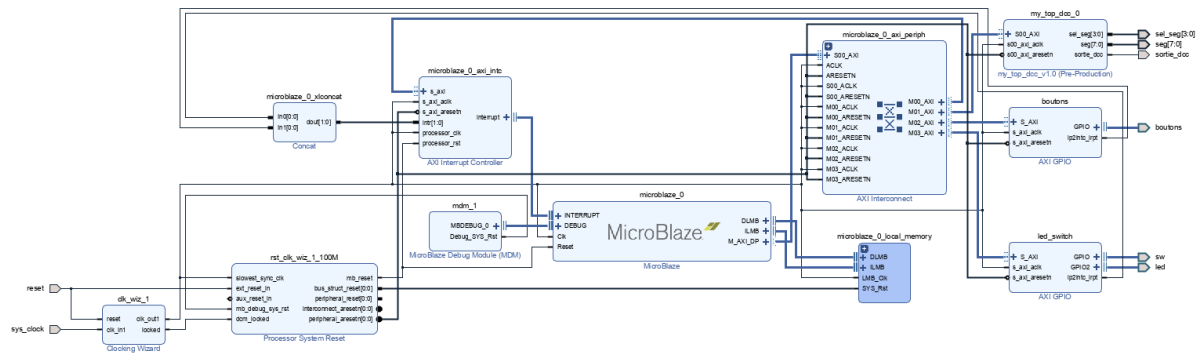


Fig. 10. Block Diagram

La **Figure 12** offre une meilleure compréhension de la hiérarchie de l'IP de la Centrale DCC appelée my_top_dcc.

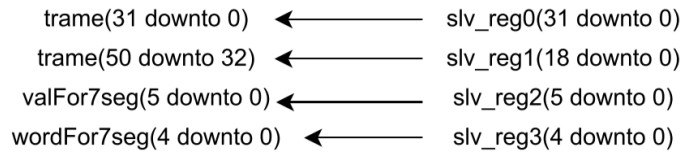


Fig. 11. Liaison avec les slave registers

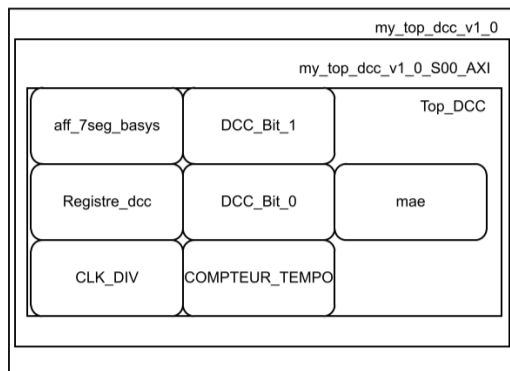


Fig. 12. Hiérarchie de l'IP

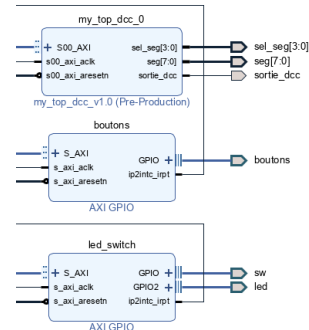


Fig. 13. Zoom sur les IP essentiels du Block Diagram

Dans le module my_top_dcc_v1_0_S00_AXI (celui qui instancie Top_DCC, voir Figure 11), la trame sur 51 bits est récupérée grâce à 2 registres. De même les paramètres permettant l'affichage sur le 7SD sont récupérés via les registres 2 et 3.

II.5 Afficheur sept segment

II.5.1 Description du module

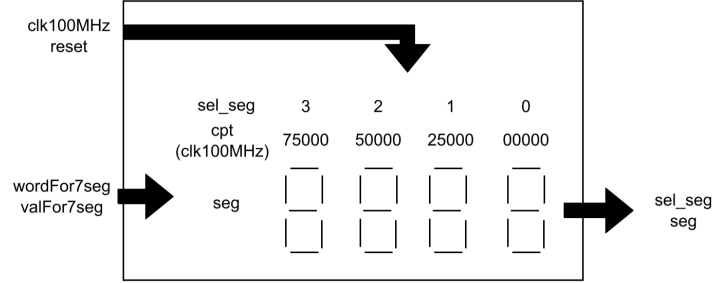


Fig. 14. Principe de l'affichage sur le 7SD

Dans ce module le message affiché sur le 7SD est configuré par 2 entrées, `wordFor7seg` et `valFor7seg`. L'entrée `wordFor7seg` reflète l'état des interrupteurs et indique le type de configuration qui est entrain d'être opéré sur la carte, donc il permet de déterminer le mot à afficher sur le 7SD. L'entrée `valFor7seg` correspond au numéro de fonction, vitesse ou adresse sur lequel s'est positionné l'utilisateur. Un compteur compte de 0 à 99999 : à 0 il affiche le caractère sur le bloc 0 (`sel_seg=0`), à 25000 sur le bloc 1, à 50000 sur le bloc 2 et à 75000 sur le bloc 3. Le compteur est remis à 0 lorsqu'il atteint 99999.

SW	valFor7seg	wordFor7seg	7SD
0	1	1	Adr1
0	2	1	Adr2
0	3	1	Adr3
0	4	1	Adr4
0	5	1	Adr5
0	6	1	Adr6

Tableau 5: Adresse du train

SW	valFor7seg	wordFor7seg	7SD
2	0	4	Sb00
2	1	4	Sb01
2	2	4	Sb02
2	...	4	SbXX
2	30	4	Sb30
2	31	4	Sb31

Tableau 7: Marche arrière

SW	valFor7seg	wordFor7seg	7SD
3	0	8	CL00
3	1	8	CL01
3	2	8	CL02
3	...	8	CLXX
3	30	8	CL30
3	31	8	CL31

Tableau 9: Désactivation des fonctions

SW	valFor7seg	wordFor7seg	7SD
1	0	2	SF00
1	1	2	SF01
1	2	2	SF02
1	...	2	SFXX
1	30	2	SF30
1	31	2	SF31

Tableau 6: Marche avant

SW	valFor7seg	wordFor7seg	7SD
3	0	8	FC00
3	1	8	FC01
3	2	8	FC02
3	...	8	FCXX
3	30	8	FC30
3	31	8	FC31

Tableau 8: Activation des fonctions

SW	valFor7seg	wordFor7seg	7SD
14	Aucun	16	RSTF

Tableau 10: Trame de reset

SW	valFor7seg	wordFor7seg	7SD
Aucun	Aucun	0	dCC

Tableau 11: Etat de repos

III Ajout de la centrale DCC comme IP dans un système Microblaze

III.1 Organigramme du code C du Microblaze

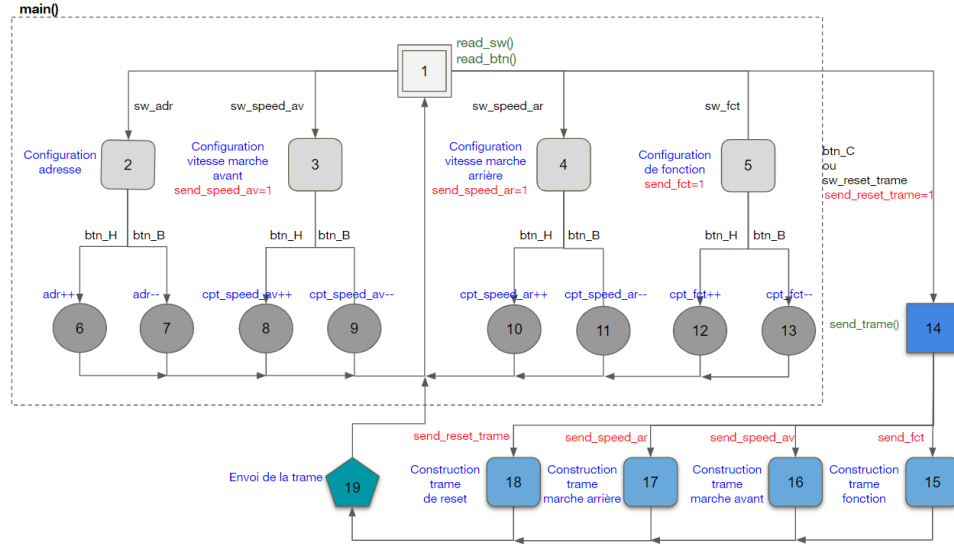


Fig. 15. Organigramme du code C

- [1] : Lecture des interrupteurs et boutons
- [2] : L'interrupteur 0 est levé, on configure l'adresse
- [3] : L'interrupteur 1 est levé, on configure une marche avant (mise à 1 de send_speed_av)
- [4] : L'interrupteur 2 est levé, on configure une marche arrière (mise à 1 de send_speed_ar)
- [5] : L'interrupteur 3 est levé, on configure une fonction (mise à 1 de send_fct)
- [6] [7] : Bouton.haut/Bouton.bas : incrémentation/décrémentation de l'adresse
- [8] [9] : Bouton.haut/Bouton.bas : incrémentation/décrémentation de la vitesse marche avant
- [10] [11] : Bouton.haut/Bouton.bas : incrémentation/décrémentation de la vitesse marche arrière
- [12] [13] : Bouton.haut/Bouton.bas : incrémentation/décrémentation de la fonction
- [14] : Appel de la fonction pour l'envoi de la trame de configuration ou de reset
- [15] : Construction de la trame de configuration de fonction si send_fct=1
- [16] : Construction de la trame de configuration de marche avant si send_speed_av=1
- [17] : Construction de la trame de configuration de marche arrière si send_speed_ar=1
- [18] : Construction de la trame de reset si send_reset_trame=1
- [19] : Envoi de la trame

III.2 Guide d'utilisation

III.2.1 Interrupteurs

Interrupteur 0 : Active à l'état haut la sélection de l'adresse

Interrupteur 1 : Active à l'état haut la sélection de la marche avant

Interrupteur 2 : Active à l'état haut la sélection de la marche arrière

Interrupteur 3 : Active à l'état haut la sélection d'une fonction

Interrupteur 14 : Génère la trame de reset

Interrupteur 15 : Reset

III.2.2 Boutons

Bouton haut : Permet d'incrémenter lorsqu'un des **Interrupteurs 0 à 3** est à l'état **haut**

Bouton bas : Permet de décrémenter lorsqu'un des **Interrupteurs 0 à 3** est à l'état **haut**

Bouton centre : Permet d'envoyer la trame lorsque les configurations sont choisies

Lorsqu'**aucun** Interrupteur n'est à l'état haut, le 7SD affiche **dCC**, cela correspond à une attente de saisie par l'utilisateur.

Afin d'envoyer une commande, vous devez :

- Choisir l'adresse du train. **Dès lors que vous choisissez une adresse et que vous ne la modifiez pas, celle-ci sera conservée pour toutes les commandes avenir.**
- Choisir le type de commande : Activation marche avant, activation marche arrière, activation/désactivation fonction ou reset de la mémoire du train.

III.2.3 Sélection de l'adresse du train

Tant que vous ne modifiez pas l'adresse, celle-ci sera conservée pour toutes les commandes avenir.

- **Levez l'interrupteur 0** (le 7SD devrait afficher AdrXX : **Adr** pour "Adresse" et XX le numéro de l'adresse)(cf **Tableau 5**)
- **Positionnez** vous sur l'adresse en **incrémentant** à l'aide du **bouton haut** ou en **décrémentant** à l'aide du **bouton bas** (elle peut être comprise entre 1 et 6).
- **Baissez l'interrupteur 0** pour enregistrer votre choix.

III.2.4 Activation de la marche avant

- **Levez l'interrupteur 1** (le 7SD devrait afficher SFXX : **SF** pour "Speed Forward" et XX le numéro de la vitesse)(cf **Tableau 6**), la numérotation correspond à celle du cahier des charges rappelé ci-dessous sur le **Tableau 12**.
- **Positionnez** vous sur la vitesse désirée à l'aide des **boutons haut** et **bas**.
- **Baissez l'interrupteur 1** afin d'enregistrer votre sélection.
- **Appuyez** sur le **bouton central** pour envoyer la trame.

III.2.5 Activation de la marche arrière

- **Levez l'interrupteur 2** (le 7SD devrait afficher SbXX : **Sb** pour "Speed backward" et XX le numéro de la vitesse)(cf **Tableau 6**), la numérotation est celle de là **Tableau 13**.
- **Positionnez** vous sur la vitesse désirée à l'aide des **boutons haut** et **bas**.
- **Baissez l'interrupteur 2** afin d'enregistrer votre sélection.
- **Appuyez** sur le **bouton central** pour envoyer la trame.

III.2.6 Activation d'une fonction

- **Levez l'interrupteur 3** (le 7SD devrait afficher FCXX : **FC** pour "Fonction" et XX le numéro de la fonction)(cf **Tableau 7**), la numérotation est celle du **Tableau 12-13**.
- **Positionnez** vous sur la fonction désirée à l'aide des **boutons haut** et **bas**.
- **Baissez l'interrupteur 3** afin d'enregistrer votre sélection.
- **Appuyez** sur le **bouton central** pour envoyer la trame.

III.2.7 Désactivation d'une fonction

- **Levez l'interrupteur 3**
- **Positionnez** vous sur la fonction à désactiver à l'aide des **boutons haut** et **bas**.
- **Baissez l'interrupteur 3** afin d'enregistrer votre sélection.
- **Appuyez** sur le **bouton central** pour envoyer la trame. Sur le 7SD vous devriez observer CLXX : **CL** pour "Clear" et XX la fonction désactivée. (cf **Tableau 14**)

III.2.8 Reset de la mémoire du train

Levez l'interrupteur 14.

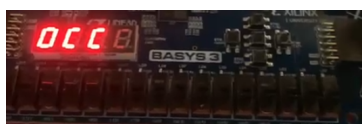


Fig. 16. Affichage DCC

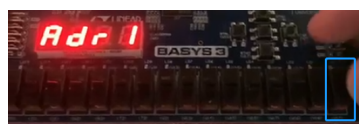


Fig. 17. Affichage adrX



Fig. 18. Affichage SFXX



Fig. 19. Affichage SBXX

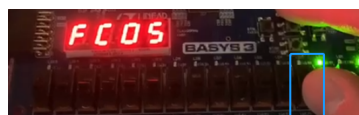


Fig. 20. Affichage FCXX

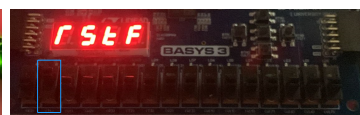


Fig. 21. Affichage rstf

Ces figures offrent un petit aperçu de l'affichage sur le 7SD.

Vitesse	Numéro
Stop	0
Stop (I)	1
E-Stop	2
E-Stop (I)	3
Step 1	4
Step 2	5
Step 3	6
Step 4	7
Step 5	8
Step 6	9
Step 7	10
Step 8	11
Step 9	12
Step 10	13
Step 11	14
Step 12	15
Step 13	16

Tableau 12: Vitesses(1)

Vitesse	Numéro
Step 14	17
Step 15	18
Step 16	19
Step 17	20
Step 18	21
Step 19	22
Step 20	23
Step 21	24
Step 22	25
Step 23	26
Step 24	26
Step 25	27
Step 26	28
Step 26	29
Step 27	30
Step 28	31

Tableau 13: Vitesses(2)

Fonction	Numéro
F0 Lumière on/off	0
F1 Son on/offon/off	1
F2 Cor Français #1	2
F3 Cor Français #2	3
F4 Turbo off	4
F5 Compresseur	5
F6 Accélération	6
F7 Courbe grincement	7
F8 Ferroviaire Clank	8
F9 Ventilateur	9
F10 Conducteur de signal	10
F11 Court Cor Français #1	11
F12 Court Cor Français #2	12
F13 L'annonce station française #1	13
F14 L'annonce station française #2	14
F15 Signal d'alerte française #1	15
F16 Signal d'alerte française #2	16
F17 Porte chauffeur ouvrir/fermer	17
F18 Valve	18
F19 Attelage	19
F20 Sable	20
F21 Libération de freins	21

Tableau 14: Fonctions

IV Bilan d'avancement et démonstration

La video suivante permet d'avoir un aperçu du résultat (Note: le 7SD a été intégré après que la vidéo ait été filmée): **Cliquer ici**

Pour ce projet, nous sommes parvenus à réaliser l'ensemble de la partie matérielle en simulant chaque module de la Centrale DCC. Puis, nous avons intégré la Centrale DCC dans un système Microblaze en tant qu'IP pour interagir avec le bus AXI et réaliser la partie logicielle. Dans la partie logicielle, notre objectif était de réaliser une interface que nous estimons pratique pour l'utilisateur. Cela inclut le fait de pouvoir choisir rapidement une commande tout en offrant un retour visuel sur le paramétrage qui est en train d'être opéré à l'aide du 7SD.