INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

PROJET VHDL

Auteurs :
KUZU Kevin
CADIOU Adrien
4EII-A

Université de Rennes

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

SOMMAIRE

SOMMAIRE **2**
INTRODUCTION **3**
Le Noeud **4**
1. Le Noeud : Entité central du système 4
2. Test bench de l'entité Noeud 6
Le Réseau **8**
1. Le Réseau 8
2. Test bench de l'entité Réseau 10
CONCLUSION **12**

INTRODUCTION

Dans le cadre de notre formation en 4ème année à l'INSA, nous avons réalisé ce projet dédié à la conception d'un réseau de communication en anneau en langage VHDL. L'enjeu principal était de transformer les concepts théoriques vus en cours en une application concrète, afin de nous approprier les spécificités du design matériel, bien différent de la programmation logicielle classique.

Le projet a été structuré autour d'un défi progressif :

- Le Nœud : Nous avons d'abord conçu et testé un "Nœud" capable de jongler entre plusieurs tâches : se tester lui-même, transmettre les messages des autres ou envoyer ses propres données.

- Le Réseau : Nous avons ensuite interconnecté ces unités pour bâtir un réseau en anneau de taille variable (N×N), capable de faire circuler l'information de manière fluide entre chaque entité.

Dans ce rapport, nous reviendrons d'abord sur la logique interne de notre entité « Node », en expliquant comment nous avons géré ses priorités de fonctionnement. Nous confronterons ensuite nos simulations aux exigences du sujet pour valider notre approche. Enfin, nous détaillerons la mise en place du réseau complet, en mettant en lumière les solutions techniques retenues pour l'adressage et la gestion des flux de données entre les nœuds.

Le Noeud

1. Le Noeud : Entité central du système

Le nœud constitue la brique élémentaire de notre architecture. Pour garantir une transmission fiable des données, son fonctionnement est entièrement synchrone, cadencé par le front montant du signal d'horloge CLK.

L'information circule sous la forme de trames de 8 bits, structurées en deux champs distincts:

- 4 bits d'adressage : dédiés à l'identification du nœud destinataire.
- 4 bits de données : transportant le message proprement dit.

Voici le récapitulatif des entrées/sorties de notre noeud :

Port	Description	Type VHDL	Direction
CLK	Horloge (front montant synchrone)	std_logic	in
TR_IN	Entrée des trames (8 bits)	std_logic_vector(7 downto 0)	in
TR_OUT	Sortie des trames (8 bits)	std_logic_vector (7 downto 0)	out
TEST_IN	Demande d'auto-test	std_logic	in
TEST_OK	Résultat d'auto-test	std_logic	out
ENV_MES	Demande d'envoi d'un message	std_logic	in
MESS_IN	Message à envoyer (4 bits)	std_logic_vector (3 downto 0)	in
MESS_OUT	Message réceptionné (4 bits)	std_logic_vector (3 downto 0)	out

Pour traduire ces spécifications, nous avons respecté le comportement interne du nœud selon une hiérarchie de priorités décroissante comme demandé dans le sujet (Auto-test, transfert de trame, ,envoi de message, default mode). Voici le schéma d'un noeud avec ses entrée/sortie ci-dessous :

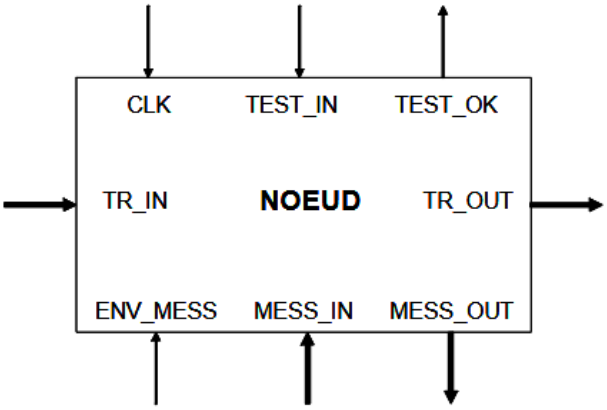

Figure 1 : Vue externe d'un nœud

Nous présentons ci-après le diagramme bloc de l'entité Node, détaillant le comportement de notre nœud configuré pour notre implémentation VHDL :

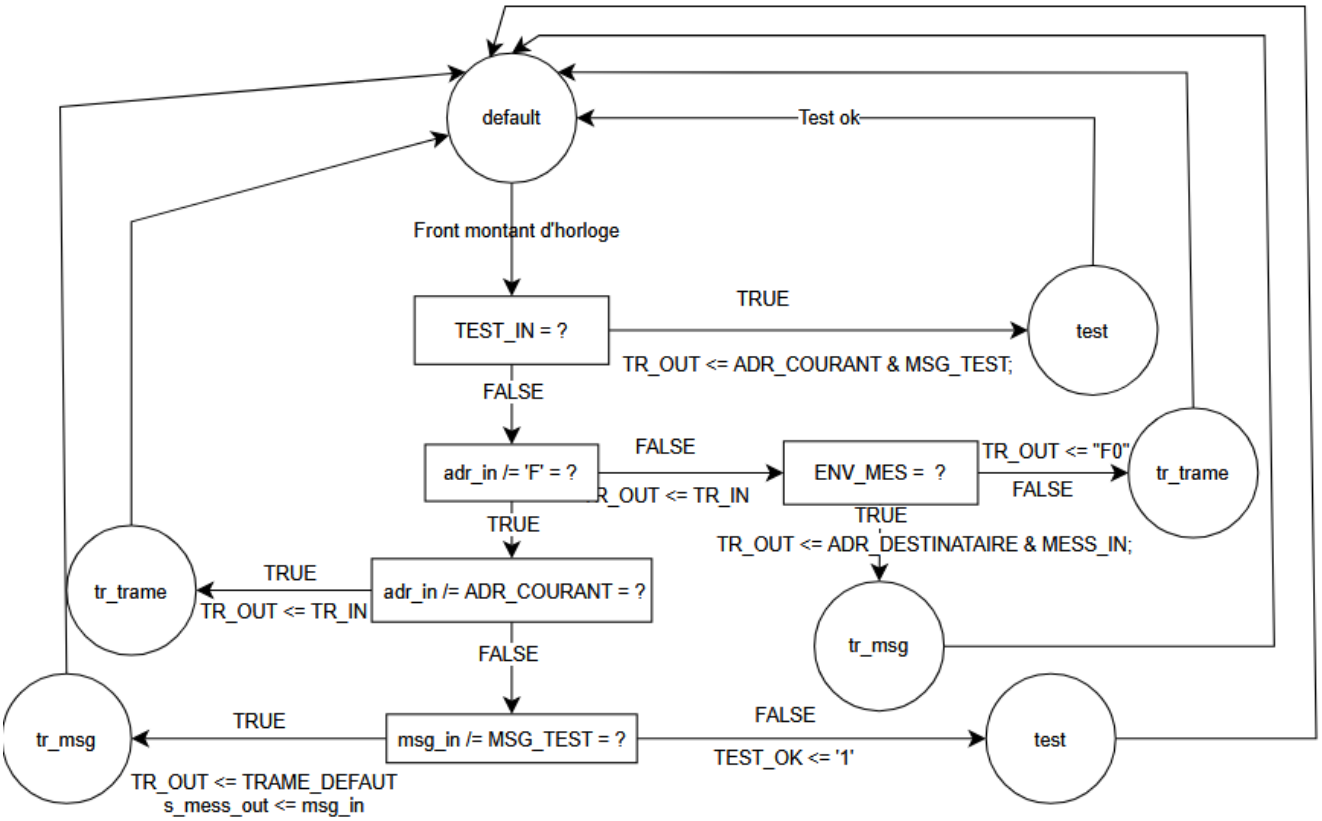

Figure 2 : Schéma bloc du comportement d'un noeud

2. Test bench de l'entité Noeud

Cette étape de simulation a pour but de valider la conception de notre entité et de garantir sa parfaite conformité avec le cahier des charges. Nous débutons notre analyse par la vérification de l'adressage, en nous assurant que les adresses courantes et de destination sont correctement générées. L'objectif de ce banc de test est de simuler l'environnement d'un nœud isolé pour vérifier son comportement face aux différents scénarios prévus par le cahier des charges. Pour cette simulation, le nœud est configuré avec les paramètres génériques suivants : Adresse courante = 5 ("0101") et Adresse destinataire = 8 ("1000").

L'architecture TEST instancie le nœud (UUT - *Unit Under Test*) et définit des signaux internes pour l'observation sur le chronogramme :

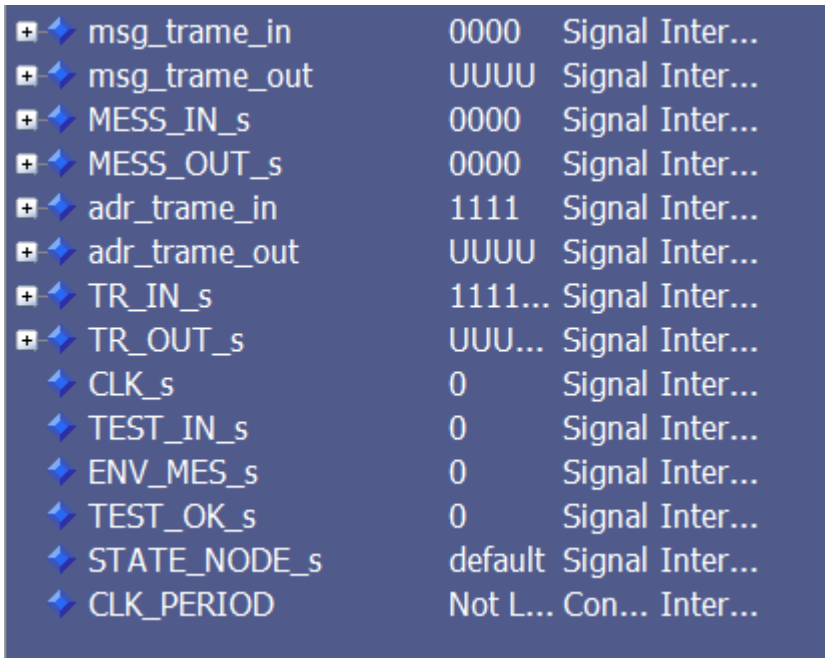

Figure 3 : Signaux affiché pour le test bench d'un noeud

- Décodage des trames : Les signaux adr_trame et msg_trame décomposent les vecteurs de 8 bits (TR_IN / TR_OUT) en champs d'adresse et de message de 4 bits pour une lecture immédiate lors de la simulation.
- Génération d'horloge : Un processus CLK_gen crée un signal périodique de 20 ns, servant de base de temps synchrone pour tout le système.

Nous avons suivi le même scénario de près ou de loin fourni dans les annexes du projet. Voici les quatre scénarios de Test :

Phase	Action de Simulation	Résultat Attendu
1 - Auto-test	TEST_IN_s <= ‘1’	Emission d’une trame x”5C”
2 - Envoi	ENV_MES_s <= ‘1’ Message = x”E”	Emission d’une trame x”8E”
3 - Recopie	Injection de x”31”	Le noeud doit recopier x”31” sur TR_OUT
4 - Récéption	Injection de x”59”	MESS_OUT doit afficher x”9” et TR_OUT doit passer à x”F0”

Voici le résultat de la simulation ci-dessous de la simulation :

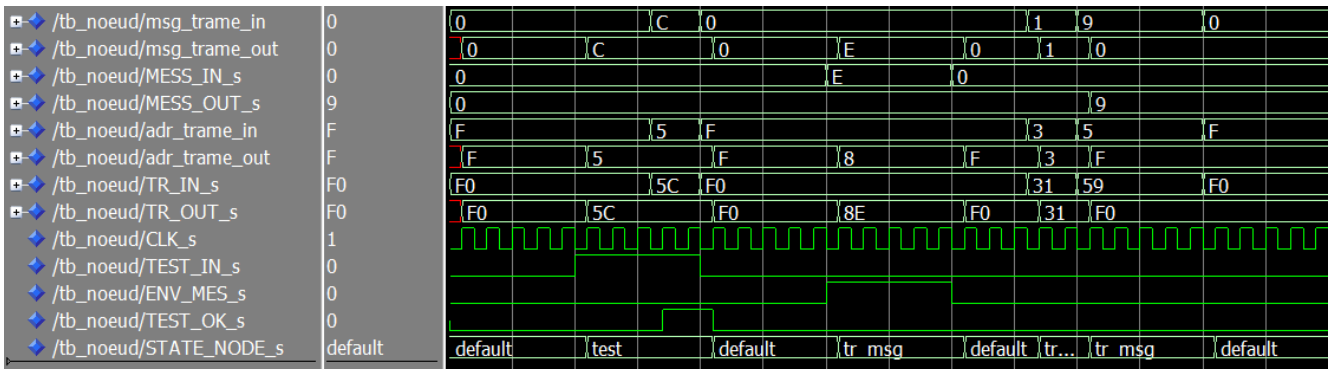

Figure 4 : Résultat de simulation de l’entité noeud sous Modelsim

Le banc de test permet de confirmer visuellement la conformité avec les chronogrammes de référence fournis dans le sujet.

- Validation de l'Auto-test : En réinjectant la trame émise (x"5C") sur l'entrée TR_IN, on vérifie que le signal TEST_OK passe bien à '1'.
- Priorité du trafic : Le test démontre qu'une trame entrante pour une autre adresse est prioritaire sur l'envoi d'un nouveau message, évitant ainsi toute collision sur le futur anneau.

Ce code de test assure que l'unité de base (NOEUD) est robuste et respecte les protocoles d'adressage définis d’après le cahier des charges. Une fois ces tests validés, le nœud peut être instancié plusieurs fois dans l'entité RESEAU pour former l'anneau final.

Le Réseau

1. Le Réseau

L'étape suivante consiste à interconnecter les entités N0EUD pour former un réseau de communication synchrone. L'architecture a été pensée pour être totalement modulaire et paramétrable.

Le réseau est défini comme une matrice carrée de taille N×N, où N est un paramètre générique compris entre 2 et 4 d'après le cahier des charges. L'interconnexion physique des nœuds suit une topologie en anneau :

- La sortie TR_OUT du nœud k est reliée à l'entrée TR_IN du nœud k+1.
- Le dernier nœud (M−1) boucle sur le premier (0) pour fermer l'anneau, assurant ainsi la circulation continue des trames.

Seuls le premier nœud (Haut-Gauche, index 0) et le dernier nœud (Bas-Droite, index N2−1) possèdent des accès vers l'extérieur du composant. Les nœuds intermédiaires fonctionnent de manière autonome en mode "relais".

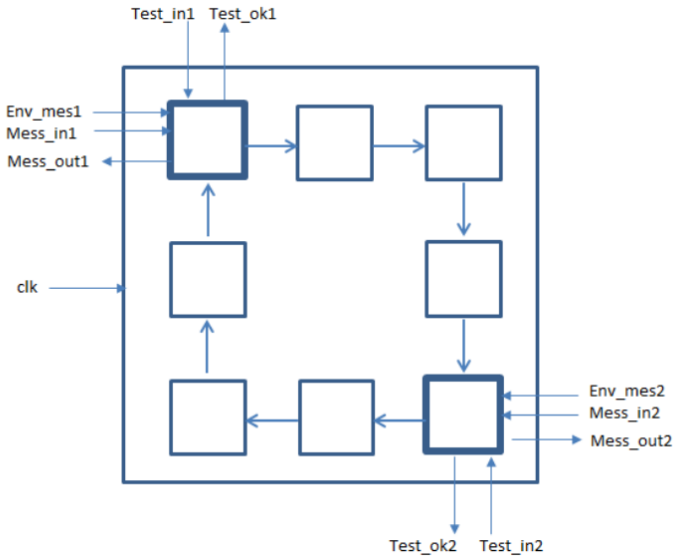

Figure 5 : Spécification du réseau en anneau

Avec comme entrées sorties :

Port	Noeud assuré	Type VHDL	Direction
CLK	Horloge (front montant synchrone)	std_logic	in
Test_in1	Haut à gauche	std_logic	in
Test_ok1	Haut à gauche	std_logic	out
Env_mes1	Haut à gauche	std_logic	in
Mess_in1	Haut à gauche	std_logic_vector (3 downto 0)	in
Mess_out1	Haut à gauche	std_logic_vector (3 downto 0)	out
Test_in2	Bas à Droite	std_logic	in
Test_ok2	Bas à Droite	std_logic	out
Env_mes2	Bas à Droite	std_logic	in
Mess_in2	Bas à Droite	std_logic_vector (3 downto 0)	in
Mess_out2	Bas à Droite	std_logic_vector (3 downto 0)	out

L'une des difficultés principales du projet réside dans l'attribution des adresses au sein de cette structure variable. Pour répondre aux exigences du cahier des charges , nous avons implémenté une matrice de constantes 2D, basée sur les spécifications du sujet.

Le code suivant définit l'ensemble des adresses disponibles pour un réseau allant jusqu'à une taille 4×4. Cette matrice est utilisée par chaque nœud pour identifier son adresse courante et celle de son destinataire en fonction de ses coordonnées (i,j).

```
-- Matrice de constante pour la gestion de l'adressage (Question 4)
constant MATRICE_ADR_COUR : ADR_MATRIX_4x4 := (
    (0 => "0000", 1 => "0001", 2 => "0011", 3 => "0100"), -- Ligne 0 (Adresses 0, 1, 3, 4)
    (0 => "1100", 1 => "0010", 2 => "0100", 3 => "0101"), -- Ligne 1 (Adresses 12, 2, 4, 5)
    (0 => "1011", 1 => "0110", 2 => "0101", 3 => "0110"), -- Ligne 2 (Adresses 11, 6, 5, 6)
    (0 => "1010", 1 => "1001", 2 => "1000", 3 => "0111")  -- Ligne 3 (Adresses 10, 9, 8, 7)
);
```

Figure 6 : Code VHDL de la Matrice d'adresse courante

Ainsi, l'instanciation du réseau repose sur la boucle GEN_NOEUDS. Ce bloc de code assure le câblage physique de l'anneau et la logique d'adressage.

Pour chaque nœud I du réseau (allant de 0 à M−1), nous calculons sa position théorique dans la grille N×N :

- I_COORD := I / N : Détermine la ligne (i) par division entière.
- J_COORD := I mod N : Détermine la colonne (j) par l'opération modulo.

Pour automatiser l'envoi de messages vers le nœud opposé, nous calculons les coordonnées du destinataire par symétrie centrale:

- I_DEST := N - 1 - I_COORD
- J_DEST := N - 1 - J_COORD

Une fois les coordonnées (i,j) connues pour la source et la destination, le code "pioche" les adresses binaires directement dans notre matrice de référence.

Et pour finir, le port map réalise le chaînage physique:

- L'entrée TR_IN du nœud actuel est branchée sur s_trame(I).
- La sortie TR_OUT est branchée sur s_trame((I + 1) mod M).

Après la phase de conception architecturale, nous passons désormais à la phase de simulation fonctionnelle. Cette étape permet de confronter notre code aux exigences du cahier des charges, notamment en observant les chronogrammes d'échanges de données entre le nœud initial et le nœud destinataire à travers les nœuds intermédiaires.

2. Test bench de l'entité Réseau

Le banc de test global a pour objectif de valider l'interconnexion des nœuds et la circulation des trames sur l'ensemble de l'anneau. Pour cette simulation, nous avons configuré le réseau avec un paramètre N=3, soit un total de 9 nœuds.

L'architecture instancie le composant `RESEAU` (UUT) et définit une période d'horloge de 10 ns. Comme le réseau comporte 9 nœuds, une trame a besoin d'au moins 9 coups d'horloge pour faire un tour complet. Le code utilise donc des temporisations de `10 * CLK_PERIOD` pour laisser le temps aux signaux de se propager.

Voici les scénarios simulé pour vérifier le bon comportement du réseau :

Phase	Action de Simulation	Résultat Attendu
1 - Auto-Test du Nœud 0 (Haut à Gauche)	Test_in1 <= ‘1’	Test_ok1 passe à ‘1’
2 - Auto-Test du Nœud 8 (Bas à Droit)	Test_in2 <= ‘1’	Test_ok2 passe à ‘1’
3 - Communication Source-Destination (Nœud 0 vers Nœud 8)	Injection d’un message x”A”	Mess_out2 renvoi x”A”

Voici le résultat de la simulation ci-dessous de la simulation :

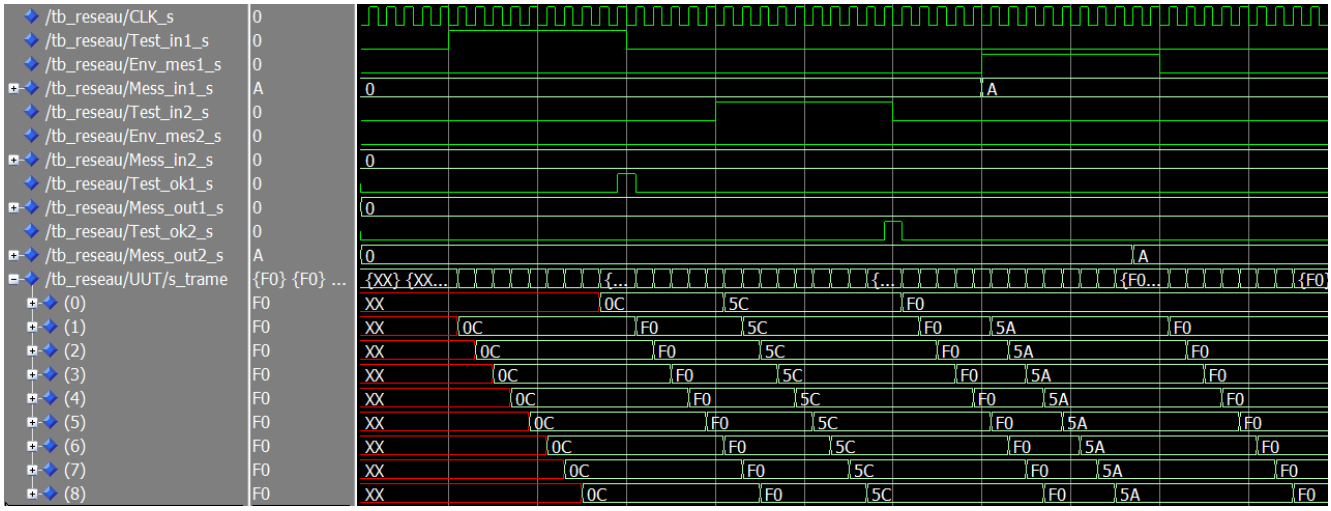

Figure 7 : Résultat de simulation de l’entité réseau pour n = 3 sous Modelsim

La simulation globale du réseau confirme la robustesse de notre architecture en anneau et la validité de notre logique d'adressage. L'analyse des chronogrammes permet de tirer les conclusions suivantes :

Nous observons le succès des tests de boucle pour le nœud "Haut-Gauche" (I=0) et le nœud "Bas-Droite" (I=M−1). Le signal Test_ok passe à l'état haut après que la trame de test a parcouru l'intégralité de l'anneau, validant ainsi la continuité physique de la chaîne.

Les messages injectés transitent de nœud en nœud à chaque front d'horloge (mode forwarding) jusqu'à l'identification de l'adresse cible. Ce comportement est conforme aux exigences de transfert unidirectionnel du cahier des charges.

Les sorties Mess_out1 et Mess_out2 affichent correctement les messages qui leur sont destinés.

CONCLUSION

Ce projet a permis de concevoir et de valider une architecture réseau complète en VHDL. Sa réalisation a favorisé l'approfondissement des notions fondamentales de la logique synchrone et la maîtrise des descriptions structurelles et comportementales.

L'utilisation de la généricité et des boucles de génération automatique s'est avérée indispensable pour répondre à l'exigence d'une taille de réseau variable tout en maintenant une structure d'adressage cohérente. En conclusion, ce travail démontre la viabilité de l'architecture proposée et souligne l'efficacité du langage VHDL pour créer des systèmes de communication modulaires et performants.

INSA Rennes
20 avenue des Buttes de Coësmes
CS 70839
35708 Rennes cedex 7
Tél : + 33 (0)2 23 23 82 00
www.insa-rennes.fr

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
Liberté
Égalité
Fraternité