

Rapport de Code VHDL Projet UAL

Rédigé par:

Ahmed Abdelhedi

Yessine Abdedayem

Fourat Boujdaria

Farouk Ben Lachheb

Ce module VHDL implémente un **verrou numérique avec un multiplicateur intégré**.

Description des fonctionnalités

1. Vérification du code secret

- Un code secret à 4 bits est comparé avec l'entrée utilisateur.
- Si le code est correct, le système se déverrouille.
- En cas d'échec répété (3 tentatives incorrectes), le verrouillage devient permanent.

2. Gestion des états du verrou

- État verrouillé** : Par défaut, le système est verrouillé.
- État déverrouillé** : L'entrée correcte du code active le déverrouillage.
- État bloqué** : Après trois tentatives erronées, le système est bloqué.

3. Multiplication de deux valeurs

- Un mode spécifique permet de multiplier deux nombres sur 4 bits.
- Le résultat est affiché sur un afficheur 7 segments.

4. Affichage et retour visuel

- L'afficheur 7 segments affiche le résultat de la multiplication.
- Des LEDs indiquent l'état du verrou (verrouillé, déverrouillé, bloqué).

5. Réinitialisation du système

- Un bouton permet de réinitialiser l'état du verrou et les tentatives.

project_lock2 - [C:/Users/User/Downloads/Atelier7p/project_lock2/project_lock2.xpr] - Vivado 2018.2

File Edit Flow Tools Repgrts Window Layout View Help Q: IO Ports write_bitstream Complete

How Navigator

- Add Sources
- Language Templates
- IP Catalog
- IP INTEGRATOR
 - Create Block Design
 - Open Block Design
 - Generate Block Design
- SIMULATION
 - Run Simulation
- RTL ANALYSIS
 - Open Elaborated Design
- SYNTHESIS
 - Run Synthesis
 - Open Synthesized Design
- IMPLEMENTATION
 - Run Implementation
 - Open Implemented Design
 - Constraints Wizard
 - Edit Timing Constraints
 - Report Timing Summary
 - Report Clock Networks

IMPLEMENTED DESIGN - xc7a35lcp236-1 (active)

Sources Netlist

DigitalLock_Multiplier

- Nets (239)
- Leaf Cells (188)

IO Ports Properties

All ports

Name: All ports

Project Summary Device x: verro.vhd x: io_constraints.xdc

C:/Users/User/Downloads/Atelier7p/project_lock2/project_lock2/srcs/consts_tlnewio_constraints.xdc

```
1 set_property IOSTANDARD LVCMOS33 [get_ports {ANODE[3]}]
2 set_property IOSTANDARD LVCMOS33 [get_ports {ANODE[2]}]
3 set_property IOSTANDARD LVCMOS33 [get_ports {ANODE[1]}]
4 set_property IOSTANDARD LVCMOS33 [get_ports {ANODE[0]}]
5 set_property IOSTANDARD LVCMOS33 [get_ports {INFOT_BITS[3]}]
6 set_property IOSTANDARD LVCMOS33 [get_ports {INFOT_BITS[2]}]
7 set_property IOSTANDARD LVCMOS33 [get_ports {INFOT_BITS[1]}]
8 set_property IOSTANDARD LVCMOS33 [get_ports {INFOT_BITS[0]}]
9 set_property IOSTANDARD LVCMOS33 [get_ports {MULTI_A[3]}]
10 set_property IOSTANDARD LVCMOS33 [get_ports {MULTI_A[2]}]
11 set_property IOSTANDARD LVCMOS33 [get_ports {MULTI_A[1]}]
12 set_property IOSTANDARD LVCMOS33 [get_ports {MULTI_A[0]}]
13 set_property IOSTANDARD LVCMOS33 [get_ports {MULTI_B[3]}]
14 set_property IOSTANDARD LVCMOS33 [get_ports {MULTI_B[2]}]
15 set_property IOSTANDARD LVCMOS33 [get_ports {MULTI_B[1]}]
16 set_property IOSTANDARD LVCMOS33 [get_ports {MULTI_B[0]}]
```

Tcl Console Messages Log Reports Design Runs IO Ports x Power DRC Methodology Timing

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	IO Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
CALCULATE	IN		W15	<input checked="" type="checkbox"/>	14	LVCMOS33*	3.300				NONE
CLK	IN		W5	<input checked="" type="checkbox"/>	34	LVCMOS33*	3.300				NONE
LOCKED	OUT		E19	<input checked="" type="checkbox"/>	14	LVCMOS33*	3.300		12	SLOW	NONE
RESET	IN		U17	<input checked="" type="checkbox"/>	14	LVCMOS33*	3.300				NONE
SEL_OP	IN		V15	<input checked="" type="checkbox"/>	14	LVCMOS33*	3.300				NONE
UNLOCKED	OUT		U16	<input checked="" type="checkbox"/>	14	LVCMOS33*	3.300		12	SLOW	NONE
VALIDATE	IN		W14	<input checked="" type="checkbox"/>	14	LVCMOS33*	3.300				NONE

Mode 1 (Sel_op=1)

Tache 1:Affichage du résultat de la multiplication sur l'afficheur 7 segments

Entrées (Inputs)

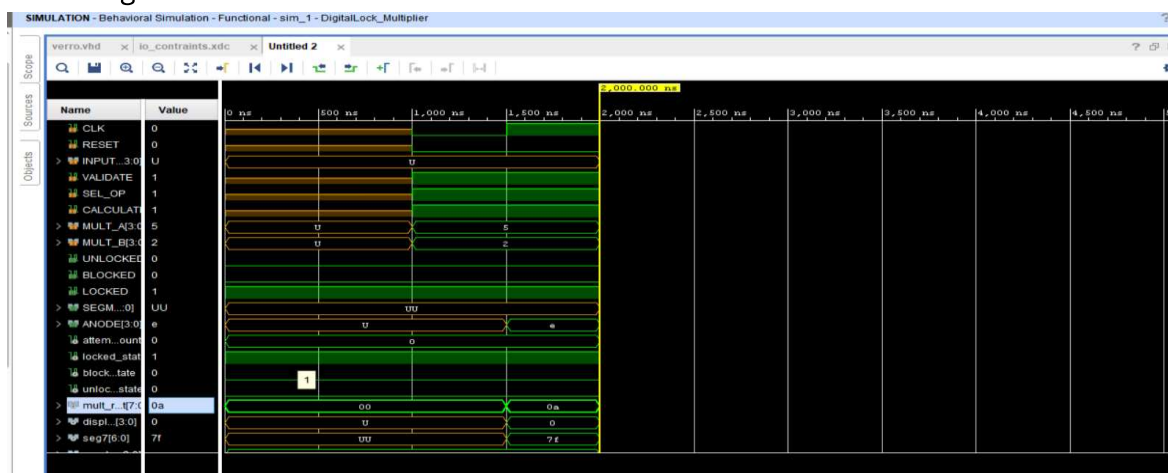
- **MULT_A** (*STD_LOGIC_VECTOR(3 downto 0)*) : Premier opérande de la multiplication.
- **MULT_B** (*STD_LOGIC_VECTOR(3 downto 0)*) : Deuxième opérande de la multiplication.
- **CALCULATE** (*STD_LOGIC*) : Signal de validation du calcul.
- **CLK** (*STD_LOGIC*) : Horloge pour synchronisation.
- **RESET** (*STD_LOGIC*) : Réinitialisation du système.

Sorties (Outputs)

- **SEGMENTS** (*STD_LOGIC_VECTOR(6 downto 0)*) : Contrôle des segments de l'affichage 7 segments.
- **ANODE** (*STD_LOGIC_VECTOR(3 downto 0)*) : Sélectionne quel afficheur est activé.

Description

Lorsqu'un utilisateur entre deux nombres binaires 4 bits (**MULT_A** et **MULT_B**) et active le signal **CALCULATE**, le système effectue la multiplication et stocke le résultat sur 8 bits. Ce résultat est ensuite converti en BCD (Binary-Coded Decimal) pour être affiché sur un afficheur 7 segments multiplexé. L'affichage alterne entre les chiffres du résultat pour permettre une lecture complète. En cas de réinitialisation (**RESET** activé), l'affichage est remis à zéro.



Tache 2: Stockage du résultat de la multiplication dans un registre

Entrées (Inputs)

- **MULT_A** (*STD_LOGIC_VECTOR(3 downto 0)*) : Premier opérande de la multiplication.
- **MULT_B** (*STD_LOGIC_VECTOR(3 downto 0)*) : Deuxième opérande de la multiplication.
- **CALCULATE** (*STD_LOGIC*) : Signal de validation du calcul.
- **CLK** (*STD_LOGIC*) : Horloge pour synchronisation.
- **RESET** (*STD_LOGIC*) : Réinitialisation du registre.

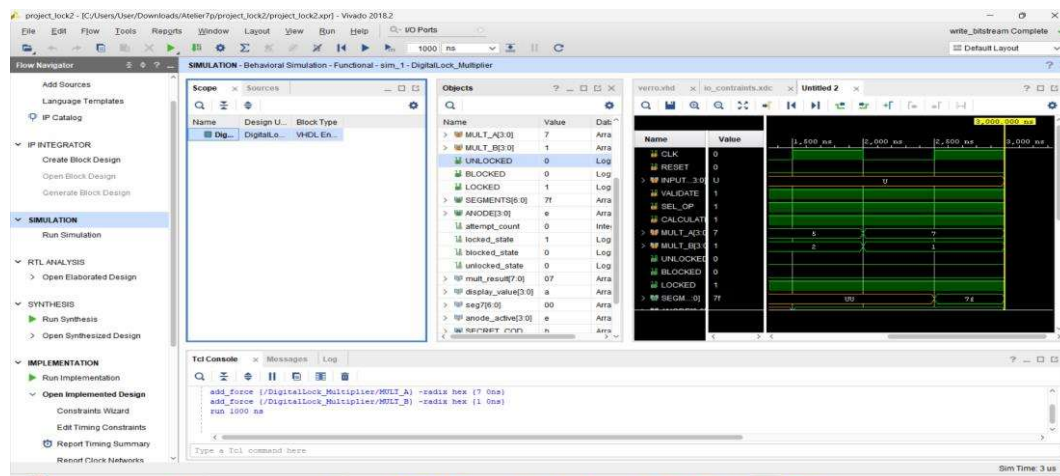
Sorties (Outputs)

- **RESULT_REG** (*STD_LOGIC_VECTOR(7 downto 0)*) : Registre stockant le résultat de la multiplication.

Description

Lorsque le signal **CALCULATE** est activé, le système multiplie les entrées **MULT_A** et **MULT_B**. Le produit (8 bits) est alors stocké dans le registre **RESULT_REG**. Ce registre conserve la valeur même après la fin de l'opération, jusqu'à ce qu'une nouvelle multiplication soit effectuée ou que le signal **RESET** soit activé, ce qui remet **RESULT_REG** à zéro.

Ce stockage permet d'afficher le résultat sur un afficheur 7 segments sans recalculer à chaque cycle d'horloge.



Mode 2 (Sel_op=0)

tache 1 : Saisie et validation du code

Entrées (Inputs)

- **CLK** (*STD_LOGIC*) : Signal d'horloge pour synchronisation.
- **RESET** (*STD_LOGIC*) : Réinitialisation du système.
- **INPUT_BITS** (*STD_LOGIC_VECTOR(3 downto 0)*) : Code saisi par l'utilisateur.
- **VALIDATE** (*STD_LOGIC*) : Signal de validation du code saisi.

Sorties (Outputs)

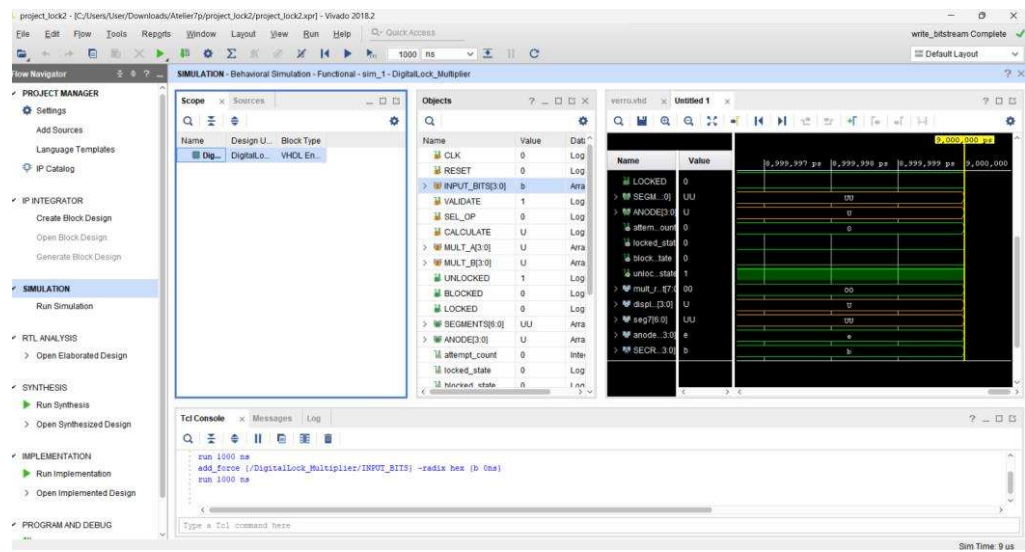
- **LOCKED** (*STD_LOGIC*) : Indique que le verrou est actif (code incorrect ou non saisi).
- **UNLOCKED** (*STD_LOGIC*) : Indique que le code est correct et que le verrou est déverrouillé.

Description

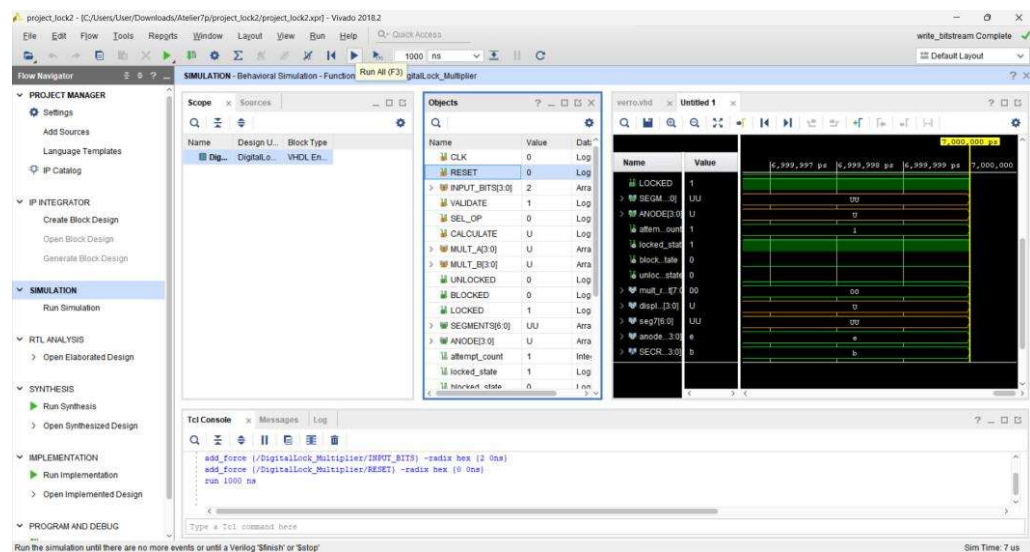
L'utilisateur entre un code via **INPUT_BITS** et valide son entrée avec **VALIDATE**.

- Si le code correspond au **SECRET_CODE**, la sortie **UNLOCKED** passe à '1', signifiant que l'accès est autorisé.
- Si le code est incorrect, la sortie **LOCKED** reste active et le verrou demeure fermé.
- Un signal **RESET** permet de réinitialiser l'état du système.

Si le code est correct, le verrou **reste verrouillé (LOCKED = '1')**



Si le code est incorrecte, le verrou **reste verrouillé (UNLOCKED = '1')**
attempt_count est incrémenté par 1



tache 2 : Gestion des tentatives et blocage du verrou (compteur)

Entrées (Inputs)

- **CLK** (*STD_LOGIC*) : Signal d'horloge pour synchronisation.
- **RESET** (*STD_LOGIC*) : Réinitialisation du système.
- **VALIDATE** (*STD_LOGIC*) : Signal de validation du code saisi.

Sorties (Outputs)

- **BLOCKED** (*STD_LOGIC*) : Indique que le système est bloqué après plusieurs tentatives incorrectes.

Description

Chaque tentative incorrecte incrémente un compteur d'erreurs.

- Si le nombre d'essais dépasse une limite définie (ex. 3 tentatives), la sortie **BLOCKED** passe à '1', empêchant toute nouvelle saisie.
- Tant que le verrou n'est pas bloqué, l'utilisateur peut réessayer de saisir le code.
- Un **RESET** permet de réinitialiser le compteur et de débloquent le système.

