

## Projet Co-design (Info 4) version du 02 Mars 2021

1. Suivez le tutorial ci-dessous pour concevoir le processeur Nios II tout en ciblant la carte DE10 Nano au lieu de la carte DE1 (donc device Cyclone V au lieu de Cyclone II).

<https://www.badprog.com/electronics-quartus-ii-creating-your-first-sopc-with-qsys-and-nios-ii-software>

### Remarques :

- Il faut utiliser la version « **Classic** » du Nios II processor
  - La taille de la mémoire on-chip (Onchip\_memory) doit être égale à **200 K Bytes**
  - N'oubliez pas de rajouter un module Jtag\_uart avec les paramètres par défaut.
2. Au niveau programmation sous Eclipse, commencez par la Template « Hello World » et tester l'exécution sur le Nios II (faite valider par l'enseignant)
  3. Avec la template « Count Binary », allumer les 8 leds en appuyant sur l'un des push-button, ensuite allumer une led sur 2 en appuyant sur un autre Push-button.
  4. Revenir dans Qsys pour rajouter un port PIO de taille 32bits configuré en Output et qui s'appelle seven\_seg.

### Remarque :

A chaque fois que vous modifier l'architecture du processeur Nios II (en rajoutant des périphériques par exemple), il faut :

- i. recalculer les adresses
  - ii. générer le processeur (generate)
  - iii. remplacer tous les fichiers sources dans Quartus (par précaution)
  - iv. Compiler votre projet Quartus et reconfigurer de nouveau votre FPGA
5. Ecrire un programme de comptage sur les 4 afficheurs 7 segments en BCD. Le programme doit compter en décimal de 0 jusqu'à 9999 tout en affichant le résultat sur les 4 afficheurs 7 segments.
  6. Revenir dans Qsys et rajoutez 4 ports PIO d'entrée de taille 4 bits chacun et qui s'appellent Unites, Dizaines, Centaines, Milliers
  7. Rajouter l'IP fréquencemètre en schéma symbole avec sorties au niveau des latches !
  8. Connecter le fréquencemètre au processeur Nios II
  9. Ecrire un programme en C qui calcule la fréquence du signal mesuré en Hz (sur la base de la valeur de N donné par le module fréquencemètre).
  10. Modifiez le programme du Nios II pour que la sélection du calibre soit faite automatiquement : la valeur de N ne doit pas dépasser 9999 et la fréquence de référence doit être toujours supérieure à la fréquence mesurée.

Tableau de brochage pour les afficheurs 7 segments

Afficheur	Segment	Repère de la broche reliée au segment	Numéro de broche sur le GPIO 1	
A	a	GPIO_1[9]	10	AH27
	a	GPIO_1[23]	26	AE22
	a	GPIO_1[34]	39	AE19
	a	GPIO_1[28]	33	AG18
B	b	GPIO_1[11]	14	AH26
	b	GPIO_1[25]	28	AF21
	b	GPIO_1[32]	37	AG15
	b	GPIO_1[30]	35	AF18
C	c	GPIO_1[7]	8	AF27
	c	GPIO_1[21]	24	AA20
	c	GPIO_1[35]	40	AE17
	c	GPIO_1[26]	31	AG19
D	d	GPIO_1[3]	4	AD26
	d	GPIO_1[17]	20	AH22
	d	GPIO_1[31]	36	AF20
	d	GPIO_1[22]	25	AF22
E	e	GPIO_1[0]	1	Y15
	e	GPIO_1[13]	16	AF25
	e	GPIO_1[27]	32	AH19
	e	GPIO_1[18]	21	AH21
F	f	GPIO_1[5]	6	AF28
	f	GPIO_1[19]	22	AG21
	f	GPIO_1[33]	38	AE20
	f	GPIO_1[24]	27	AG20
G	g	GPIO_1[1]	2	AC24
	g	GPIO_1[15]	18	AF23
	g	GPIO_1[29]	34	AH18
	g	GPIO_1[20]	23	AH23

# Copie d'écran de Qsys pour vérifier surtout les connections

Qsys - de1\_blinker.qsys (D:\Enseignements\2019\_2020\UT\INFO4(Tp)\sysonprogchip\de1\_blinker.qsys)

File Edit System Generate View Tools Help

IP Catalog

- Project
- Library
- System
- Basic Functions
- DSP
- Interface Protocols
- Low Power
- Memory Interfaces and Controllers
- Processors and Peripherals
- Qsys Interconnect
- University Program

System Contents

System: de1\_blinker Path: nios2\_proc

Connections	Name	Description	Export	Clock	Base	End	Tags
✓	clock_main	Clock Source	clk	export			
✓	clk_in	Clock Input	clk				
✓	clk_in_reset	Reset Input	clk				
✓	clk_reset	Reset Output	clk				
✓	onchip_memory	On-Chip Memory (RAM or ROM)					
✓	clk1	Clock Input	clk...		0x8_0000	0xb_0d3f	
✓	s1	Avalon Memory Mapped Slave	clk1				
✓	reset1	Reset Input	clk1				
✓	switcher	PIO (Parallel I/O)					
✓	clk	Clock Input	clk...		0x50	0x5f	
✓	reset	Reset Input	clk				
✓	s1	Avalon Memory Mapped Slave	clk				
✓	external_connection	Conduit	switcher_e...				
✓	clk	Clock Input	clk...		0x60	0x6f	
✓	reset	Reset Input	clk				
✓	s1	Avalon Memory Mapped Slave	clk				
✓	external_connection	Conduit	led_externa...				
✓	sysid_1659	System ID Peripheral					
✓	clk	Clock Input	clk...		0x70	0x77	
✓	reset	Reset Input	clk				
✓	control_slave	Avalon Memory Mapped Slave	clk				
✓	nios2_proc	Nios II (Classic) Processor					
✓	clk	Clock Input	clk...				
✓	reset_n	Reset Input	clk				
✓	data_master	Avalon Memory Mapped Master	clk				
✓	instruction_master	Avalon Memory Mapped Master	clk				
✓	d_irq	Interrupt Receiver	clk				
✓	jtag_debug_module_reset	Reset Output	clk				
✓	jtag_debug_module	Avalon Memory Mapped Slave	clk		0x1000	0x17ff	
✓	custom_instruction_master	Custom Instruction Master	clk				
✓	jtag_uart	JTAG UART					
✓	clk	Clock Input	clk...		0x78	0x7f	
✓	reset	Reset Input	clk				
✓	avalon_jtag_slave	Avalon Memory Mapped Slave	clk				
✓	seven_seg	PIO (Parallel I/O)					
✓	clk	Clock Input	clk...		0x40	0x4f	
✓	reset	Reset Input	clk				
✓	s1	Avalon Memory Mapped Slave	clk				
✓	external_connection	Conduit	seven_seg...				
✓	clk	Clock Input	clk...		0x30	0x3f	
✓	reset	Reset Input	clk				
✓	s1	Avalon Memory Mapped Slave	clk				
✓	external_connection	Conduit	unites_exte...				
✓	clk	Clock Input	clk...		0x0	0xf	
✓	reset	Reset Input	clk				
✓	s1	Avalon Memory Mapped Slave	clk				
✓	external_connection	Conduit	dizaines_ex...				
✓	clk	Clock Input	clk...		0x20	0x2f	
✓	reset	Reset Input	clk				
✓	s1	Avalon Memory Mapped Slave	clk				
✓	external_connection	Conduit	centaines_e...				
✓	clk	Clock Input	clk...		0x10	0x1f	
✓	reset	Reset Input	clk				
✓	s1	Avalon Memory Mapped Slave	clk				

Hierarchy

- de1\_blinker (de1\_blinker.qsys)
- centaines external connection
- clk
- dizaines external connection
- led external connection
- milliers external connection
- seven\_seg external connection
- switcher external connection
- unites external connection
- centaines
- dizaines
- milliers
- jtag\_uart
- clock\_main
- led
- sysid\_1659
- onchip\_memory
- seven\_seg
- switcher
- connections

Parameters

System: de1\_blinker Path: nios2\_proc

Nios II (Classic) Processor  
altera\_nios2\_qsys

Core Nios II Caches and Memory Interfaces Advanced Features MMU and MPU Set

Select a Nios II Core

Nios II Core:  Nios II/e  Nios II/s  Nios II/f

Nios II Selector Guide	Nios II/e	Nios II/s	Nios II/f
RISC 32-bit	RISC 32-bit	RISC 32-bit	RISC 32-bit
Instruction Cache	Instruction Cache	Instruction Cache	Instruction Cache
Branch Prediction	Branch Prediction	Branch Prediction	Branch Prediction
Hardware Multiply	Hardware Multiply	Hardware Multiply	Hardware Multiply
Hardware Divide	Hardware Divide	Hardware Divide	Hardware Divide
Dynamically Reconfigurable	Dynamically Reconfigurable	Dynamically Reconfigurable	Dynamically Reconfigurable

Memory Usage (e.g. Stratix IV) Two M9Ks (or equiv.) Two M9Ks + cache Three M9Ks

Hardware Arithmetic Operation

Hardware multiplication type: Embedded Multipliers

Hardware divide:

Reset Vector

Reset vector memory: onchip\_memory.s1

Reset vector offset: 0x00000000

Reset vector: 0x00080000

Exception Vector

Exception vector memory: onchip\_memory.s1

Exception vector offset: 0x00000020

Exception vector: 0x00080020

MMU and MPU

Include MMU

Only include the MMU using an operating system that explicitly supports an MMU

Fast TLB Miss Exception vector memory: None

Fast TLB Miss Exception vector offset: 0x00000000

Fast TLB Miss Exception vector: 0x00000000

Include MPU

0 Errors, 1 Warning

Generate HDL... Finish