

Rapport de TP - ESN11 - Lab 1

Davy MILLION
M2 SETSIS

19 janvier 2023

1 Introduction

Ce *Lab* nous propose de prendre en main la mise en place d'un système-sur-puce programmable (*SoPC*) sur cible Intel Max 10, via l'outil de synthèse / *SoC Builder* Quartus. Pour ce faire, l'objectif est d'implémenter un *led chaser* (chenillard).

2 Développements

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
<input checked="" type="checkbox"/>		clk_0	Clock Source					
		clk_in	Clock Input	clk	exported			
		clk_in_reset	Reset Input	reset				
		clk	Clock Output	Double-click to export	clk_0			
		clk_reset	Reset Output	Double-click to export				
<input checked="" type="checkbox"/>		nios2_gen2_0	Nios II Processor					
		clk	Clock Input	Double-click to export	clk_0			
		reset	Reset Input	Double-click to export	[clk]			
		data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
		irq	Interrupt Receiver	Double-click to export	[clk]			
		debug_reset_request	Reset Output	Double-click to export	[clk]			
		debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]			
		custom_instruction_master	Custom Instruction Master	Double-click to export	[clk]			
<input checked="" type="checkbox"/>		jtag_uart_0	JTAG UART Intel FPGA IP					
		clk	Clock Input	Double-click to export	clk_0			
		reset	Reset Input	Double-click to export	[clk]			
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]			
		irq	Interrupt Sender	Double-click to export	[clk]			
<input checked="" type="checkbox"/>		prio_0	PIO (Parallel I/O) Intel FPGA IP					
		clk	Clock Input	Double-click to export	clk_0			
		reset	Reset Input	Double-click to export	[clk]			
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]			
		external_connection	Conduit	Double-click to export	prio_0_external_con...			
<input checked="" type="checkbox"/>		onchip_memory2_0	On-Chip Memory (RAM or ROM)...					
		clk1	Clock Input	Double-click to export	clk_0			
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]			
		reset1	Reset Input	Double-click to export	[clk1]			
<input checked="" type="checkbox"/>		prio_1	PIO (Parallel I/O) Intel FPGA IP					
		clk	Clock Input	Double-click to export	clk_0			
		reset	Reset Input	Double-click to export	[clk]			
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]			
		external_connection	Conduit	Double-click to export	prio_1_external_con...			
		irq	Interrupt Sender	Double-click to export	[clk]			

FIGURE 1 – Disposition interne des différents composants du *SoPC*

Nous retrouvons :

- le CPU Nios II ("*cheap version*") et sa *on-chip memory* associée (d'une taille de 40KB) ;
- 2 blocs *Parallel I/O* permettant de récupérer d'interfacer via du *soft*, pour l'un les entrées saisies sur les *switchs* (orienté en entrée du système) et pour l'autre, de piloter les LEDs (en sortie du système).

On pourra noter que le bus d'instruction ne devrait pas être relié à autre chose que la *on-chip memory*. Ceci est une erreur.

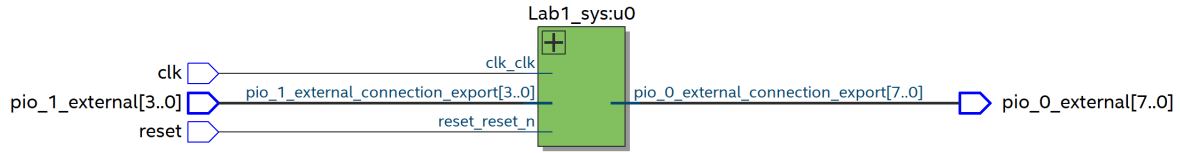


FIGURE 2 – Schéma bloc du système QSYS avec entrées/sorties

L'implémentation du chenillard a été réalisée via *polling* et interruption, cette dernière étant plus intéressante de façon générale en embarqué (asynchronisme des événements). On pourra noter que les interruptions ont été seulement utilisées pour actualiser la variable de vitesse : il aurait tout à fait été possible de cadencer la fréquence de rafraichissement du chenillard sur un compteur, dont on aurait fait varier la période. Dans ce cas, la logique de mise à jour de l'état aurait pu être elle aussi confié à une interruption.

```
while (1) {
    IOWR_ALTERA_AVALON_PIO_DATA(PIO_0_BASE, state);
    // might be a bit overkill...
    state = (state << 1) | (state >> (sizeof(state)*8 - 1));
    usleep(1400000 - (speed*90750));
}
```

Listing 1: Extrait du `main` : logique d'actualisation de l'état du chenillard

Le chenillard a été implémenté suivant la logique d'une rotation (gauche) d'un registre 8-bit.

3 Conclusion

Ce *lab* nous a permis de prendre en main la conception d'un *SoPC* sous QSYS et de bien comprendre les interfaces entre la partie *soft* et le *gateway*.