

Présentation BE VHDL

Réalisation d'une FFT 8bit en VHDL

Maraninchi Guillaume

Micha Nicolas

Encadrant : JEGO Christophe



Vitis HLS

Mise en place du protocole AXI4_Stream

Conception d'une classe pour faire la FFT

Regret de l'utilisation de la classe

Compréhension des limites de Vitis HLS

Adaptation de la classe en fonctions sans pointeurs

Vitis HLS

Synthèse à 10 et 20 ns de notre ip FFT

Résultat obtenue

Avec 10 ns → slack négative : -1.12 ns

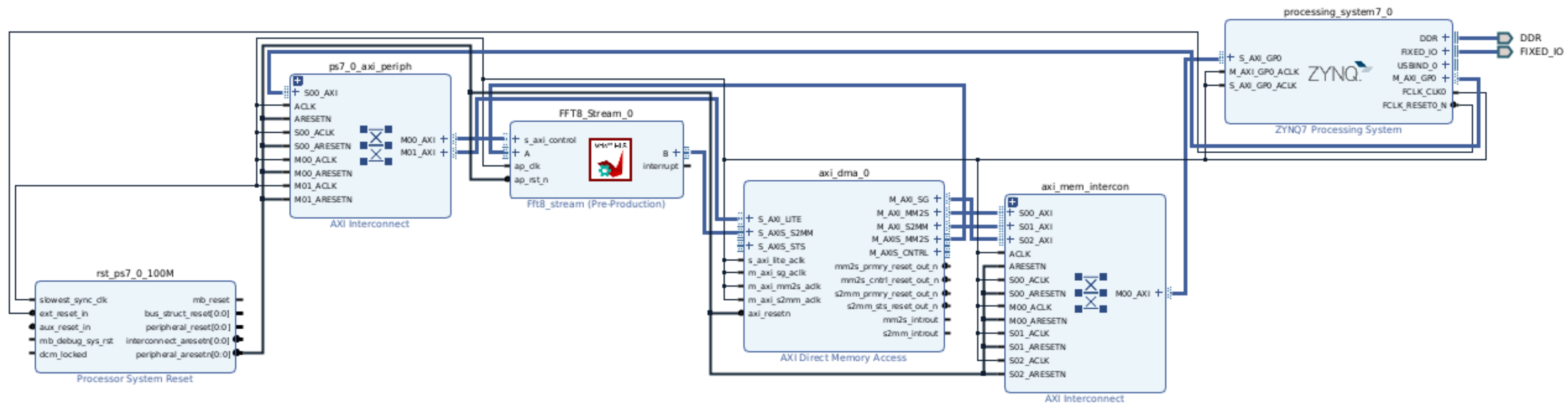
Avec 20 ns :

Target	Estimated	Uncertainty
20.00 ns	18.895 ns	1.00 ns

Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
457	9.140E3	-	458	-	no	16	166 8089	16854	0	
444	8.880E3	-	444	-	no	16	166 5593	15872	0	

Vivado

Création du bloc design pour notre IP



2. Vivado

Vivado

Synthèse du design

Erreur du aux nombre de DSP trop utilisé 224/220

Modification du calcul de W en passant par des conditions

On passe maintenant à 16/220

Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
33	660.000	-	34	-	no	0	8 3302	2484		0
20	400.000	-	20	-	no	0	8 806	1502		0

Python

Implémentation sur la carte PynQ

Ecriture du test de notre FFT avec des difficultés lié au DMA

On simplifie le DMA dans le bloc design pour faire un nouveau test

Malheureusement nous n'avons pas pu faire marcher le code de test