

# Lab 5 - Synthetiseren van de pipelined mMIPS processor

Het doel van dit lab is kennis te vergaren en ervaring op te doen door studenten over/met:

- de structuur en werking van de pipelined mMIPS processor,
- het modelleren van de pipelined mMIPS hardware in SystemC taal (RTL-niveau),
- de synthese van de pipelined mMIPS processor,
- de synthese programma's: CoCentric SystemC Compiler en Xilinx ISE,
- het optimaliseren van de mMIPS implementatie,
- het testen van de pipelined mMIPS processor.

Daarvoor wordt het synthetiseerbare SystemC model gebruikt van de pipelined mini MIPS processor (**mMIPS**) en de software beschreven in "Software mMIPS-lab".

Tijdens dit lab gaan de studenten het SystemC model van de mMIPS synthetiseren en de structuur en werking van de mMIPS analyseren en aanpassen. De mMIPS is behandeld tijdens het college en beschreven in het boek "Computer Organization & Design: The Hardware / Software Interface" van D.A. Patterson en J.L. Hennessy. Het is de bedoeling, dat de studenten dit boek tijdens dit lab bij zich hebben en het (zo nodig) kunnen raadplegen.

## Opdrachten

Dit lab bestaat uit twee opdrachten. In de eerste opdracht ga je kennismaken met het synthese traject dat gebruikt wordt om de mMIPS in de FPGA te implementeren. In de tweede opdracht ga je het ontwerp van de mMIPS optimaliseren. Het doel is om de klok-frequentie van de mMIPS te verhogen. Alleen de **tweede** opdracht moet worden afgetekend.

Voordat je aan de opdrachten kan beginnen moet je nog een aantal programma's installeren. Volg hiervoor de installatie handleiding "Toolflow mMIPS-lab Setup" op [http://www.es.ele.tue.nl/education/5JJ55-65/mmips-lab/labs/5/install\\_toolflow.php](http://www.es.ele.tue.nl/education/5JJ55-65/mmips-lab/labs/5/install_toolflow.php).

In deze opdracht wordt verwezen naar een aantal documenten waarin de werking van de verschillende software programma's wordt beschreven. Je kunt deze documenten vinden op <http://www.es.ele.tue.nl/education/Computation/mmips-lab/labs/5/toolflow.php>

### 1. Synthetiseren van de mMIPS

Het SystemC model van de mMIPS is synthetiseerbaar. Dit betekent dat het mogelijk is om dit model van de mMIPS met behulp van een aantal software programma's te implementeren in een FPGA. In deze opdracht ga je kennis maken met deze software programma's.

1. Het document "design flow" beschrijft alle software programma's die gebruikt worden om de mMIPS te simuleren op de computer en om deze te implementeren in een FPGA. Dit document beschrijft onder andere de software programma's die gebruikt worden om het SystemC model van de mMIPS te synthetiseren. Deze synthese gebeurt met behulp van *CoCentric SystemC Compiler*, *Xilinx XST* en *Xilinx ISE*. Volg de stappen die beschreven worden in de onderdelen "CoCentric SystemC Compiler", "Xilinx XST" en "Xilinx ISE" om het SystemC model van de mMIPS te synthetiseren. Het resultaat is een bit file die met behulp van het programma "celoxica-transfer" in de FPGA geïmplementeerd kan worden. Daarnaast produceert het programma "Xilinx XST" ook een rapport met de naam *mmips.srp*. Hierin wordt aangegeven hoeveel logica er nodig is om de mMIPS te implementeren. In dit rapport vind je ook informatie over het kritische pad en de maximale klokfrequentie van de mMIPS.

2. Bestudeer het rapport *mmips.srp* dat door “Xilinx XST” is gegenereerd. Je moet hierbij zoeken naar de antwoorden op de volgende vragen:
  - (a) Hoeveel logica is er nodig om de mMIPS te implementeren? De belangrijkste factor in de hoeveelheid logica die gebruikt wordt in de mMIPS is het aantal slices dat gebruikt wordt. Daarnaast spelen andere componenten zoals het aantal block rams (geheugens) en het aantal look-up tables natuurlijk ook een rol.
  - (b) Wat is de maximale klokfrequentie van de mMIPS?
  - (c) Door welke modules van de mMIPS loopt het kritische pad?

## 2. Verkorten van het kritische pad

In de vorige opdracht heb je het kritische pad van de mMIPS bestudeert. In deze opdracht ga je proberen om dit kritische pad te verkorten.

1. Bedenk een oplossing (aanpassing van het data pad) om het kritische pad te verkorten.
2. Implementeer deze oplossingen in het SystemC model van de mMIPS. Zorg er voor dat de mMIPS correct blijkt werken.
3. Bepaal met Xilinx XST hoeveel logica er nodig is voor je nieuwe oplossing. Bekijk ook de klokfrequentie en het kritische pad van je nieuwe mMIPS. Bedenk een manier om dit pad verder te verkorten.