

1. Übung: PROL16 auf dem DE1

Name(n):

Punkte:

In dieser Übung sollen Sie Ihren PROL16-Entwurf für eine FPGA Implementierung auf dem DE1 modifizieren, wobei die internen Block-RAMs des FPGAs verwendet werden sollen. Ein entsprechender Assembler, der eine fertige Block-RAM-Beschreibung erzeugt, ist in `/eda/mhe3/bin` verfügbar. Er muss mit den Parametern

```
assembler.rb -i infile.asm -o outfile.vhd
```

gestartet werden. Beachten Sie, dass Sie zur Ausführung des Assemblers einen funktionierenden Ruby-Interpreter (<http://www.ruby-lang.org>) benötigen. Außerdem muss die Speichergröße in der *.vhd Datei auf die am FPGA verfügbare Größe reduziert werden.

1 Peripherie

Entwickeln Sie zwei Peripherieeinheiten für den PROL16 die auf verschiedenen Adressbereichen angesprochen werden können:

- General Purpose Input-Port (GPI), zum Einlesen der zehn Schalterwerte (mit Spike Filter)
- General Purpose Output-Port (GPO) zur Ansteuerung der vier 7-Segment Anzeigen

Die Busschnittstelle dieser Komponenten können Sie wie jene des erzeugten RAMs gestalten.

Zur Anbindung an die CPU benötigen Sie zusätzlich einen einfachen Adressdecoder (Abbildung 1).

Hinweis zur Speicheranbindung

Der PROL16 verwendet eine asynchrone Speicherschnittstelle, an der nun ein On-Chip-Bus verwendet werden soll. Diese Schnittstelle erwartet beim Lesen nach dem Anlegen der Adressen und Strobes noch im gleichen Zyklus die entsprechenden Daten.

Mit den in den FPGAs verfügbaren Block-RAMs ist dies nicht möglich, da entweder an den Adresseingängen oder den Datenausgängen eine Registerbank modelliert werden muss [Alt08]. Die Verknüpfung der Ausgänge `mem_oe_no` und `mem_wr_no` mit dem negierten Takt wird dann nicht mehr benötigt. Für einen On-Chip-Bus ist es auch sinnvoll, aus den Speicherzugriffssignalen einen *Read*- und einen *Write-Strobe* zu erzeugen.

2 Simulation

Erstellen Sie ein Assemblerprogramm, das die 10 Schalterstellungen auf dem DE1 als binär kodierte Zahl interpretiert (Zahlenbereich 0 bis 1023) und den entsprechenden dezimalen Wert auf den vier 7-Segment Anzeigen ausgibt.

Entwickeln Sie eine Testbench mit der Sie die Funktion Ihres Designs verifizieren können. Erzeugen Sie eine *.sof Datei und lassen Sie Ihr Design auf dem DE1 laufen.

Literatur

[Alt08] Altera. *Quartus II Version 8.0 Handbook, Volume 1: Design and Synthesis*, 2008.

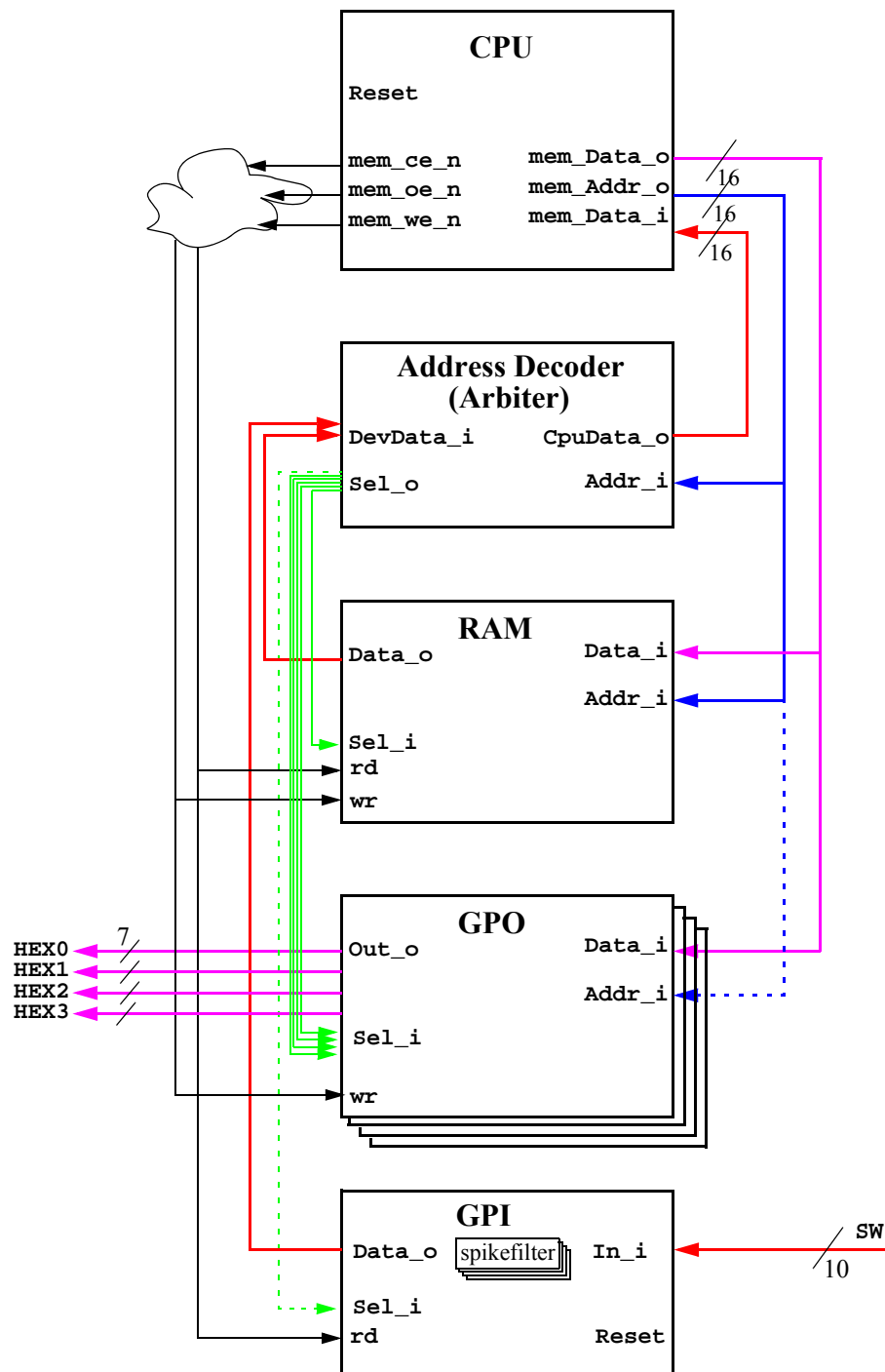


Abbildung 1: Übersicht der benötigten Einheiten und der On-Chip-Busse