

Übung 7

In dieser Übung erstellen wir den Programmzähler, sowie das Befehlsregister und einen Bus-treiber/-empfänger zur Steuerung der Kommunikation auf dem Datenbus. Im Folgenden sind die Komponenten beschrieben. Erstellen sie zu jeder Komponente eine Testbench, welche die korrekte Funktion derselben dokumentiert.

Programmzähler (PC)

Der Programmzähler arbeitet, je nach Belegung des **MODE**-Signals, als Zähler oder übernimmt eine Sprungadresse aus einem der Register (bei Sprungbefehlen).

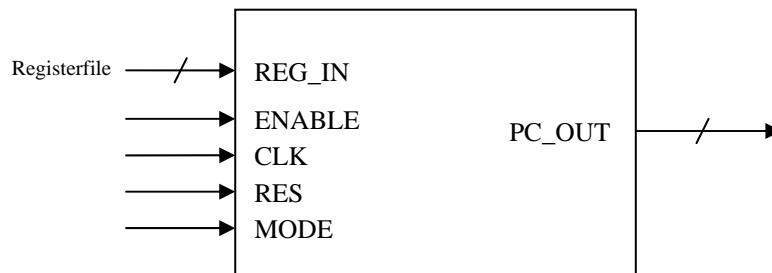


Abbildung 15: Programmzähler

Entwickeln Sie eine Verhaltensbeschreibung für den Programmzähler (*pc.vhd*, **entity pc**). Tabelle 8 zeigt die Schnittstellendefinition des Programmzählers:

Port	Richtung	Bedeutung
REG_IN(15:0)	IN	Sprungadresse aus einem Universalregister
MODE	IN	Betriebsart: MODE='0' lädt die Sprungadresse von REG_IN, MODE='1' erhöht die aktuelle Sprungadresse.
ENABLE	IN	Enable-Signal: Nur bei ENABLE='1' können Sprungadressen geladen oder der Programmzähler inkrementiert werden.
RES	IN	Reset. Bei RES='1' wird der Programmzähler initialisiert.
CLK	IN	Taktsignal
PC_OUT(15:0)	OUT	Ausgabe des Programmzählers

Tabelle 8: Portbeschreibung für den Programmzähler

Befehlsregister (IR)

Das Befehlsregister dient als Buffer für den auszuführenden Befehl. Das Befehlswort liefert zudem die Registeradressen und dient somit auch als Steuerwort für das Registerfile und den Multiplexer. Tabelle 9 dokumentiert die Schnittstelle der **entity IR** (*ir.vhd*):

Port	Richtung	Bedeutung
CLK	IN	Taktsignal
D(15:0)	IN	Dateneingang
ENABLE	IN	Enable. Nur bei ENABLE='1' D in das Register übernommen.
RES	IN	Reset. Bei RES='1' wird das Befehlsregister initialisiert.
Q(15:0)	OUT	Ausgang. Gibt den Inhalt des Registers als Steuerwort für das Registerfile und den Multiplexer aus.

Tabelle 9: Portbeschreibung des Befehlsregisters

Bustreiber/-empfänger

Der Bustreiber/-empfänger (**entity BUSIO**, *busio.vhd*) dient zur Steuerung der Ein- und Ausgabe auf den externen Datenbus.

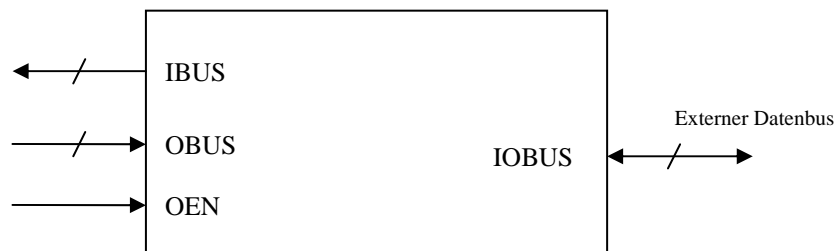


Abbildung 16: Bustreiber/-empfänger

<i>Port</i>	<i>Richtung</i>	<i>Bedeutung</i>
IOBUS(15:0)	INOUT	Bus
IBUS(15:0)	OUT	Lesen vom Bus
OBUS(15:0)	IN	Schreiben auf den Bus
OEN	IN	Bei OEN='1' wird IOBUS von OBUS getrieben, sonst wird nur gelesen

Tabelle 10: Portbeschreibung des Bustreiber/-empfängers

Erstellen Sie eine Verhaltensbeschreibung für die Komponente. Beachten Sie dabei folgendes: Es kann immer jeweils nur eine Komponente (Prozessor, Speicher,...) den Datenbus treiben, jedoch kann jede Komponente (auch die gerade treibende) vom Datenbus lesen. Vergessen Sie deshalb nicht, im „nur Lesemodus“ (OEN/= '1') den Port IOBUS auf nicht treibend (= 'Z') zu setzen.

Multiplexer

Da der Dateneingang des Registerfiles mit Daten von verschiedenen Quellen versorgt werden kann, benötigt man zur Auswahl einen Multiplexer. Die Daten kommen entweder vom Speicher (LD), vom Befehlsregister (LDIL und LDIH), von der ALU (Arithmetische Befehle) oder vom Programmzähler (CALL). Zur Umsetzung eignet sich ein 4-zu-1-Multiplexer für 16-Bit Worte (**entity MUX4x1_16**). Erstellen Sie also eine synthetisierbare Verhaltensbeschreibung des Multiplexers in VHDL.