

Übung 9

In dieser Übung kehren wir wieder zum Mikroprozessor zurück. Der Aufbau des Gesamtsystems wird vorbereitet und mit dem Entwurf des Steuerwerkes wird begonnen.

Vorbereitung zum Aufbau des Gesamtsystems

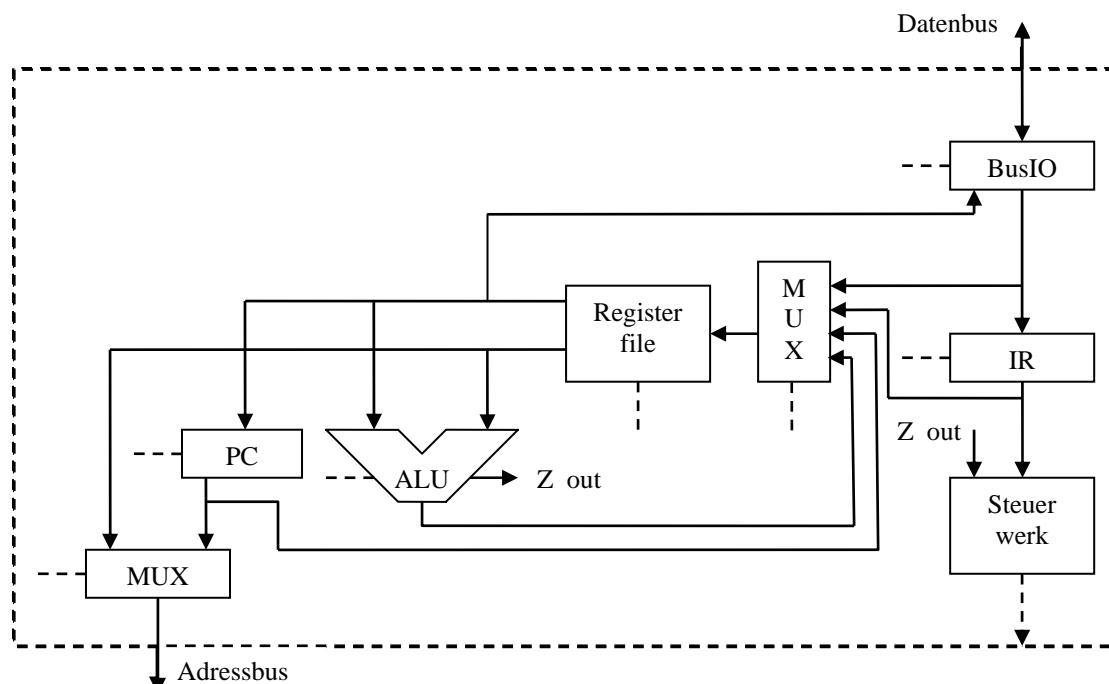
Mit den bisher entwickelten Modulen sind die Elemente für den Datenpfad entworfen. Zusammen mit dem Steuerwerk kann damit das Gesamtsystem mit folgender Schnittstelle aufgebaut werden:

Port	Richtung	Bedeutung
DATA(15:0)	INOUT	Daten vom/zum Speicher
READY	IN	Steuersignal vom Speicher
RES	IN	Reset
CLK	IN	Taktsignal
ADR(15:0)	OUT	Adresse für den Speicher
DIR	OUT	Signalisiert, dass Datenbus getrieben wird
RD	OUT	Lese-Anfrage (Read-Request)
WR	OUT	Schreib-Anfrage (Write-Request)

Tabelle 12: Portbeschreibung des Prozessors

Der bidirektionale Anschluss DATA dient zum Datenaustausch mit dem Hauptspeicher.

- Erstellen Sie für den Prozessor ein Blockdiagramm mit der **entity PROC**.
- Platzen Sie alle Komponenten des Datenpfades (ALU, Registerfile, Programmzähler, Multiplexer,...)
- Überlegen Sie sich anhand des Befehlssatzes die nötigen Verbindungen zwischen den einzelnen Komponenten
- Verbinden Sie die Komponenten untereinander, lassen Sie jedoch Steuer und Statussignale zunächst offen.



<i>Befehl</i>	<i>Opcode</i>	<i>Bedeutung</i>
LD	01 0-- xxx --- zzz --	$R_{xxx} \leftarrow M[R_{zzz}]$
ST	01 1-- --- yyy zzz --	$M[R_{zzz}] \leftarrow R_{yyy}$
LDIH	10 0-- xxx yyy zzz cc	$R_{xxx}[15:8] \leftarrow yyyzzzcc$
LDIL	10 1-- xxx yyy zzz cc	$R_{xxx}[7:0] \leftarrow yyyzzzcc$
JMP	11 00- --- yyy --- --	$pc \leftarrow R_{yyy}$
JZ	11 01- --- yyy zzz --	$pc \leftarrow R_{yyy}$ if $R_{zzz} == 0$
CALL	11 11- xxx yyy --- --	$R_{xxx} \leftarrow pc+1, pc \leftarrow R_{yyy}$
NOP	11 10- --- --- --- --	Keine Operation
ADD	00 000 xxx yyy zzz --	$R_{xxx} \leftarrow R_{yyy} + R_{zzz}$
SUB	00 001 xxx yyy zzz --	$R_{xxx} \leftarrow R_{yyy} - R_{zzz}$
AND	00 010 xxx yyy zzz --	$R_{xxx} \leftarrow R_{yyy} \text{ AND } R_{zzz}$
OR	00 011 xxx yyy zzz --	$R_{xxx} \leftarrow R_{yyy} \text{ OR } R_{zzz}$
CP	00 100 xxx --- zzz --	$R_{xxx} \leftarrow R_{zzz}$
NOT	00 101 xxx --- zzz --	$R_{xxx} \leftarrow \text{NOT } R_{zzz}$
SAL	00 110 xxx --- zzz --	$R_{xxx} \leftarrow R_{zzz} \ll 1$
SAR	00 111 xxx --- zzz --	$R_{xxx} \leftarrow R_{zzz} \gg 1, R_{xxx}[15] \leftarrow R_{zzz}[15]$

Spezifikation des Steuerwerkes

- Erstellen Sie eine Liste aller Status- und Steuersignale, die in Ihrem bisher vorhandenen Teilentwurf als Ein- und Ausgaben für das Steuerwerk benötigt werden.
- Beschreiben Sie den Befehlszyklus für alle ALU Befehle sowie für das Laden von Konstanten mit einem Zustandsübergangsdiagramm. Kommentieren Sie die einzelnen Übergänge.
- Geben Sie für jeden Befehl an, welche Steuersignale bei den entsprechenden Zuständen bzw. Zustandsübergängen gesetzt werden müssen.

Die Kommunikation mit dem Speicher erfolgt über die Signale RD, WR und READY. Bei einer Schreib- bzw. Leseanforderung durch die Signale WR bzw. RD erfolgt nach einer gewissen Verzögerungszeit die Meldung vom Speicher, dass das Datum übernommen wurde bzw. auf dem Datenbus liegt ($\text{READY} = '1'$). Das READY-Signal wird vom Speicher erst auf 0 zurückgesetzt, nachdem die Schreib- bzw. Leseanforderung zurückgesetzt worden ist. Achten Sie dabei immer auf eine korrekte Ansteuerung des Datenbusses über das DIR-Signal: $\text{DIR} = '1'$ bedeutet, dass der Prozessor den Datenbus treibt, bei $\text{DIR} = '0'$ kann der Speicher Daten auf den Bus schicken.

- Fügen Sie Ihrem Design das Testspeichermodule (testmem16.vhd) von der Homepage hinzu.
- Überlegen Sie sich anhand des Quelltextes noch einmal die Funktion des Speichermoduls
- Erstellen Sie ein neues Blockdiagramm (TestSystem.bde), das ein Testspeichermodule und ein Prozessormodule miteinander verbindet. Dieses Blockdiagramm stellt die Testumgebung für das noch zu entwickelnde Steuerwerk dar.

Entwurf des Steuerwerkes als Finite State Machine

Entwerfen Sie aus den Daten für die Spezifikation des Steuerwerkes eine FSM. ActiveHDL bietet die Möglichkeit Zustandsübergangsdiagramme grafisch einzugeben und die dazugehörige FSM als VHDL-Datei zu generieren. Gehen Sie dazu wie folgt vor:

- Mit *Add new File->State Diagram* fügen sie ihrem Design eine neue Datei (ControlUnit.asf) mit der **entity** CU hinzu. Tabelle 13 zeigt die dazugehörige Port-Belegung (Wizard benutzen):

<i>Port</i>	<i>Richtung</i>	<i>Beschreibung</i>
CLK	IN	Takt
RES	IN	Reset bei RES='1'
READY	IN	Statussignal vom Speicher
Z_OUT	IN	Statussignal von ALU
STW(...:0)	IN	Die Bits 15 bis 11 eines jeden Befehls geben den jeweiligen OpCode an
S(...:0)	OUT	Steuersignale für Registerfile, Programmzähler, Lese-, Schreibzugriffe, Befehlsregister, Multiplexer am Adressbus

Tabelle 13: Portbeschreibung des Steuerwerks

- Geben Sie vorerst nur die ALU-Befehle ein.
- Generieren Sie die VHDL-Beschreibung und erzeugen Sie ein Symbol für das Steuerwerk
- Fügen Sie das Steuerwerk in Ihren bisherigen Entwurf des Prozessors ein.
- Testen Sie den Prozessor, indem Sie den Testspeicher (TestMem16.vhd) mit ALU-Befehlen beschreiben, übersetzen, und anschließend in der Simulation die Befehle und deren Abarbeitung am Datenpfad verfolgen.
- Korrigieren Sie gegebenenfalls den bisherigen Entwurf
- Erweitern Sie das Steuerwerk schrittweise um Lade-, Speicher- und Sprungbefehle und verfahren Sie analog zu den ALU-Befehlen.