

Übungsblatt 4

Entwurf des Einzyklen-Datenpfads

Abgabefrist: Mittwoch 09.05.2018, 10:00 Uhr

1.1. Einführung

In dieser Übung werden Sie einen Einzyklen-Datenpfad entwerfen, der unter anderem die Registerbank und die ALU aus der Übung 3 nachnutzt. Ihre Aufgabe ist es, die erforderlichen VHDL-Quellen des Top-Moduls zu erstellen, diese zu strukturieren und zu simulieren.

1.2. Projektbeschreibung

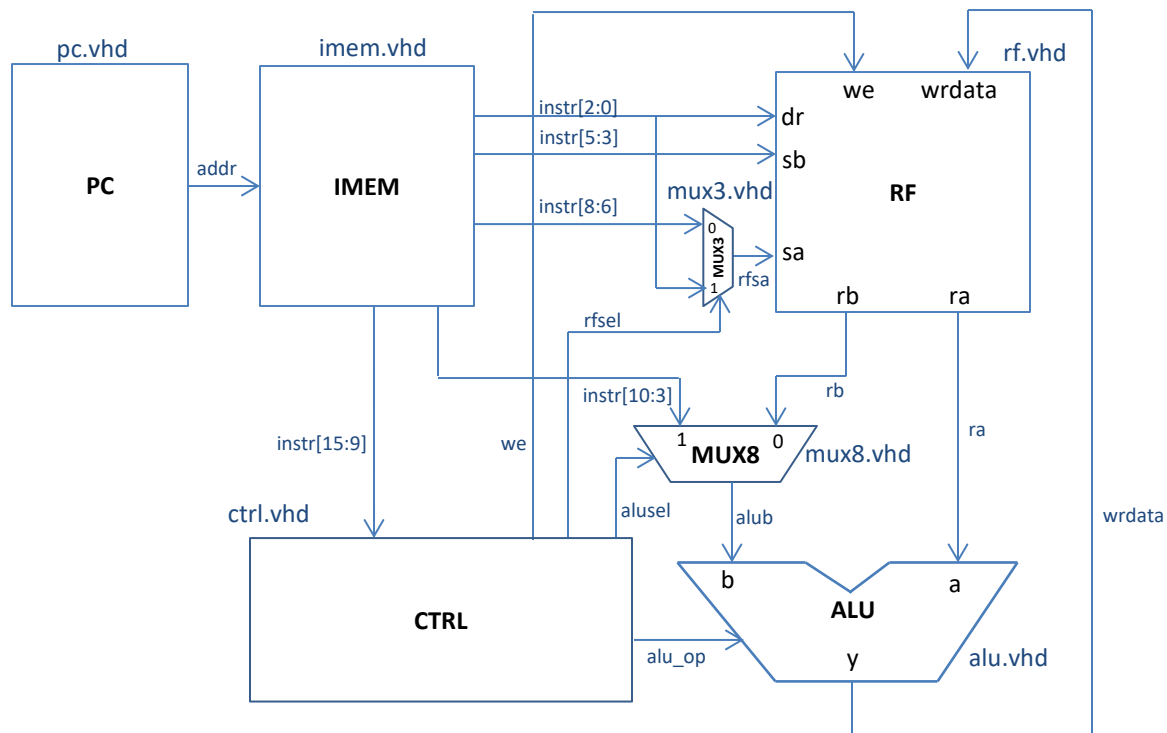


Abbildung 1.1. Einzyklen-Datenpfad Modul.

Das Blockdiagramm der zu erstellenden Datenpfad-Architektur ist in die Abbildung 1.1. dargestellt. Das Datenpfad-Modul enthält folgende Blöcke:

- Programmzähler (PC) ist ein 8-Bit-Register, dessen Wert mit jedem Taktzyklus erhöht wird. Der Anfangswert des PCs ist x"00".
- Befehlsspeicher (IMEM) ist ein 16-Bit breiter asynchroner Speicher, der das Programm enthält. Das VHDL-Verhaltensmodell des IMEM mit Testprogramm wird Ihnen online bereitgestellt.
- Registerbank (RF) wird aus Übung 3 nachgenutzt.
- Arithmetik und Logikeinheit (ALU) wird aus Übung 3 nachgenutzt.
- 8-bit-Datenmultiplexer (MUX8) ist ein Multiplexer zur Auswahl der Dateneingabe für die ALU. In Abhängigkeit vom Befehlstyp, werden die Daten aus der Registerbank (R-Typ) oder eine Konstante (I-Typ) zur ALU geführt.
- 3-bit Datenmultiplexer (MUX3) ist ein Multiplexer zur Steuerung des *sa* Eingangsports der Registerbank.
- Steuerlogik (CTRL) ist ein kombinatorischer Logikblock, der die Instruktionen dekodiert und die Steuersignale für die Multiplexer, die ALU und die Registerbank in Abhängigkeit vom Befehlstyp erstellt.

Der Datenpfad unterstützt zwei Arten von Befehlsformaten: R-Typ (Register-Typ) und I-Typ (Immediate-Typ).

Das Befehlsformat vom R-Typ ist definiert als:

type (2 bit)	opcode (5 bit)	sr1 (3 bit)	sr2 (3 bit)	dst (3 bit)
--------------	----------------	-------------	-------------	-------------

Das Befehlsformat vom I-Typ ist definiert als:

type (2 bit)	opcode (3 bit)	constant (8 bit)	sr1/dst (3 bit)
--------------	----------------	------------------	-----------------

Die Instruktioncodierung ist in der Tabelle 1 angegeben.

R-Instruction [15:0]	type [15:14]	opcode [13:11]	opcode [10:9]	sr1 [8:6]	sr2 [5:3]	dst [2:0]	Operation
add r1, r2, r3	01	000	00	r1	r2	r3	r3 = r1 + r2
sub r1, r2, r3	01	001	00	r1	r2	r3	r3 = r1 - r2
inc r1	01	010	00	r1	x	r1	r1 = r1 + 1
dec r1	01	011	00	r1	x	r1	r1 = r1 - 1
and r1,r2,r3	01	100	00	r1	r2	r3	r3 = r1 and r2
or r1,r2,r3	01	101	00	r1	r2	r3	r3 = r1 or r2
xor r1,r2,r3	01	110	00	r1	r2	r3	r3 = r1 xor r2
not r1,r3	01	111	00	r1	x	r3	r3 = not r1

I-Instruction [15:0]	Type [15:14]	opcode [13:11]	constant [10:3]			sr1/dst [2:0]	Operation
addi imm,r3	10	000	imm[7:6]	imm[5:3]	imm[2:0]	r3/r3	r3 = r3 + imm
subi imm,r3	10	001	imm[7:6]	imm[5:3]	imm[2:0]	r3/r3	r3 = r3 – imm
andi imm,r3	10	100	imm[7:6]	imm[5:3]	imm[2:0]	r3/r3	r3 = r3 and imm
ori imm,r3	10	101	imm[7:6]	imm[5:3]	imm[2:0]	r3/r3	r3 = r3 or imm
xori imm,r3	10	110	imm[7:6]	imm[5:3]	imm[2:0]	r3/r3	r3 = r3 xor imm

Tabelle 1. Instruktioncodierung.

1.3. Erstellen der Datenpfadblöcke in VHDL

Entwerfen Sie die VHDL-Quellen für die Bausteine des Datenpfads, wie in Abbildung 1.1 dargestellt. Gehen Sie wie folgt vor:

1. Erstellen Sie ein 8-Bit PC in *pc.vhd*. Der PC wird durch Reset mit ,0' initialisiert.
2. Erstellen Sie einen 2-to-1 8-bit Multiplexer in *mux8.vhd*.
3. Erstellen Sie einen 2-to-1 3-bit Multiplexer in *mux3.vhd*.
4. Erstellen Sie ein CTRL-Modul *ctrl.vhd* gemäß der Spezifikation aus Tabelle 1. Die Steuerlogik soll in einem einzigen kombinatorischen VHDL Prozess beschrieben werden!

1.4. Erstellen des Datenpfad-Top-Moduls in VHDL

1. Erstellen Sie ein strukturelles VHDL Top-Level-Design des Datenpfad-Moduls *datapath.vhd*. Importieren Sie das IMEM Speichermodell (*imem.vhd*) in Ihr Projekt. Instanzieren Sie IMEM in Ihrem Top-Modul. Importieren Sie die VHDL-Quellen für die ALU und Registerbank aus Übung 3. Stellen Sie sicher, dass 'Copy sources into projects' in dem 'Add Sources' Wizard ausgewählt ist.

Das Top-Design enthält folgende Ein- und Ausgangsports (behalten Sie die Reihenfolge bei):

- *clk* – Takteingang.
 - *rst_n* – asynchrones Reset-Signal (low aktiv).
 - *pc_data[7:0]* – Wert des PC (wird von der Testbench benutzt).
 - *instr[15:0]* – Aus IMEM gelesene Instruktion (wird von der Testbench benutzt).
 - *alu_data[7:0]* – Ausgabewert der ALU (wird von der Testbench benutzt).
 - *ex_we* – Wert des Write-Enable-Signals für die Registerbank (wird von der Testbench benutzt).
2. Führen Sie RTL-Analyse vom Top-Modul durch.

1.5. Simulation des Top-Designs

Importieren Sie die VHDL-Testbench-Datei *datapath_tb.vhd* in Ihr Projekt und simulieren Sie die Testbench mit Ihrer Datenpfadimplementierung. Der Test beinhaltet folgendes:

```
addi 0, r0          -- r0 = 0
addi 1, r1          -- r1 = 1
addi 2, r2          -- r2 = 2
addi 3, r3          -- r3 = 3
addi 4, r4          -- r4 = 4
addi 5, r5          -- r5 = 5
addi 6, r6          -- r6 = 6
addi 7, r7          -- r7 = 7
add r0, r1, r7      -- r7 = r0 + r1 = 0 + 1 = 1
sub r5, r4, r6      -- r6 = r5 - r4 = 5 - 4 = 1
inc r0              -- r0 = r0 + 1 = 0 + 1 = 1
dec r2              -- r2 = r2 - r1 = 2 - 1 = 1
and r1, r5, r5      -- r5 = r1 and r5 = x"01" and x"03" = x"01"
xori 5, r4          -- r4 = r4 xor x"05" = x"04" xor x"05" = x"01"
andi 1, r3          -- r3 = r3 and x"01" = x"03" and x"01" = x"01"
or r1, r1, r1       -- r1 = r1 or r1 = x"01" or x"01" = x"01"
addi x"FE", r0      -- r0 = r0 + x"FE" = x"01" + x"FE" = X"FF"
subi 1, r0          -- r0 = r0 - x"01" = x"FF" - x"01" = x"FE"
not r0, r0          -- r0 = not r0 = not x"FE" = x"01"
```

Die Instruktionen wurden in der VHDL-Quelle des IMEMs vorinstalliert. Überprüfen Sie die Meldungen in der 'Tcl-Konsole', um zu sehen, ob das Programm korrekt ausgeführt wird. Wenn der Test abgeschlossen ist, sollten alle Register in der Registerbank mit x"01" geladen sein. Welche Simulationszeit können Sie ablesen?

1.6. Synthese des Top-Designs

Führen Sie die Synthese des Top-Designs in Vivado durch. Generieren Sie den 'Timing Summary Report' und überprüfen Sie die Ergebnisse für 'Setup Delay' unter 'Unconstrained Paths'. Welches 'Worst Setup Delay' können Sie ablesen?