Hochschule Konstanz
Fakultät Informatik

Prof. Dr.-Ing. Michael Blaich
Robotik und Künstliche Intelligenz

Übung Rechnerarchitekturen AIN 2 SoSe2025

3. Single Cycle CPU

Die Abgabe erfolgt durch Hochladen der Lösung in Moodle. Zusätzlich wird die Lösung in der Übung nach dem Abgabetermin stichprobenartig kontrolliert.

Bearbeitung in Zweier-Teams

Team-Mitglied 1: Alexander Engelhardt

Team-Mitglied 2: Timothy Drexler

Aufgabe 3.1 Dekodierung und Ausführung einer Instruktion

In Abbildung 1 sind der Datenpfad und einige Steuerleitungen einer einfachen Eintakt-Implementierung der MIPS CPU dargestellt. Die Abbildung zeigt ebenfalls den Inhalt des Instruktionsspeichers. Der Inhalt der Register ist in Tabelle 1 gegeben.

3.1.1 Bestimmen Sie die Instruktion, die im nächsten Takt ausgeführt wird, wenn im Register `$pc` der Wert 8 steht und geben Sie den zugehörigen Befehl in Assembler an.

~~Da der PC = 8 ist, bedeutet dies, dass wir 8 Bytes bzw. 2 Words benutzen => ist die Instruktions-~~
adresse 8 / 4 => Die № 2

3.1.2 In Abbildung 1 sind einige Signale des Datenpfads mit Nummern in Kreisen versehen. Tragen Sie in der linken Hälfte von Tabelle 2 die Werte ein, die diese Signale bei der Ausführung der Instruktion im nächsten Takt annehmen.

3.1.3 Tragen Sie in der rechten Hälfte von Tabelle 2 die Werte ein, die die Steuersignale bei der Ausführung der Instruktion im nächsten Takt annehmen. Geben Sie eine kurze, stichwortartige Erklärung an, welche Unterscheidungen mit den jeweiligen Steuersignalen getroffen werden.

3.1:
PC = 8 Bytes
Instruction = PC / 4 = 2
2:
100011|10010|01110|00000|00101|100100
35 | 18 | 14 | 4 + 32 + 64 + 256 = 356
lw $s2| $t6| 356

lw $t6 356($s2);

3:
ALUSrc: 1. wenn Konstante, 0, wenn Register.
ALU Operation: Unterscheidet, welche ALU-Operation durchgeführt wird, bei lw Operation := 2 = 0010.
PCSrc: 0 bei normaler Instruktion, 1 bei Branch.
MemWrite: 1, wenn in Speicher geschrieben wird.
MemtoReg: 1, wenn das Ergebnis aus dem Speicher kommt.
MemRead: 1, wenn aus Speicher gelesen wird.
RegWrite: 1, wenn in Register geschrieben wird.

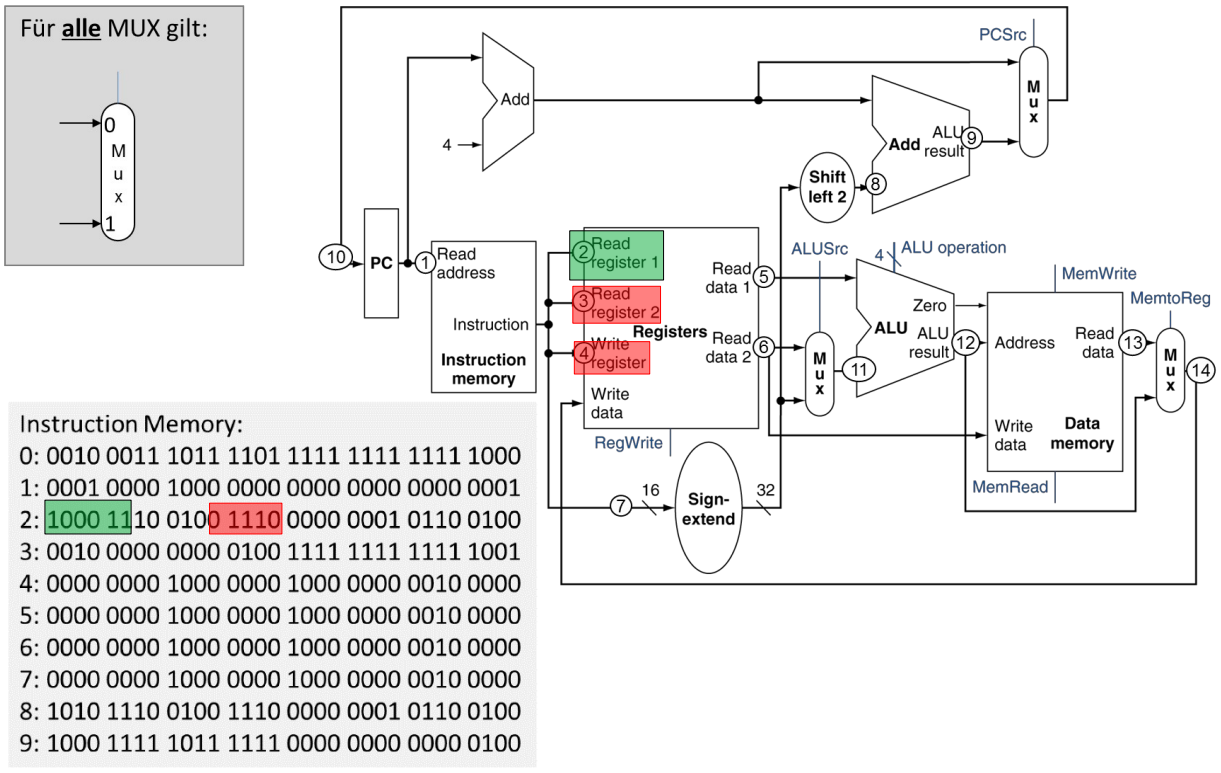

Abbildung 1: Vereinfachte Single Cycle CPU ohne Steuerwerk

0	0x0	4	0x8	8	0x15	12	0x6	16	0x2	20	0x0	24	0x0	28	0x100
1	0x3	5	0x2	9	0x17	13	0x0	17	0x3	21	0x0	25	0x0	28	0x800
2	0x4	6	0x8	10	0x0	14	0x200	18	0x80	22	0x0	26	0x0	30	0x200
3	0xA	7	0x0	11	0x0	15	0x0	19	0x0	23	0x0	27	0xFF	31	0x10

Tabelle 1: Registerinhalte in Hexadezimaldarstellung

Signale des Datenpfads (nummeriert)				Steuersignale	
1	8	8	1424 (356 << 2)	ALUSrc	1
2	10010 = 18	9	1436	ALU operation	0010
3	01110 = 14	10	12	PCSrc	0
4	01110 = 14	11	356	MemWrite	0
5	0x80	12	356 +0x80 = 484	MemtoReg	0
6	0x200	13	484	MemRead	1
7	356	14	484	RegWrite	1

Tabelle 2: Ausführung der Instruktion

Aufgabe 3.2 Jump and Link

Mit der Instruktion `jal LABEL` wird eine Prozedur aufgerufen. Format und Funktionsweise der `jal` Instruktion sind ähnlich zur jump (`j`) Instruktion. Allerdings ist der OPCODE 3 und die Instruktion speichert zusätzlich zum Sprung die Rücksprungadresse, also die Adresse der nächsten Instruktion (PC+4), im Register `$ra`. Erläutern Sie anhand einer Skizze, welche Erweiterungen Sie in Datapath und Control der idealisierten MIPS CPU vornehmen müssten, um die Instruktion `jal` zu realisieren.

Description:	Jumps to the calculated address and stores the return address in $31
Operation:	`$31 = PC + 4; PC = (PC & 0xf0000000) | (target << 2)`
Syntax:	`jal target`
Encoding:	`0000 11ii iiii iiii iiii iiii iiii iiii`

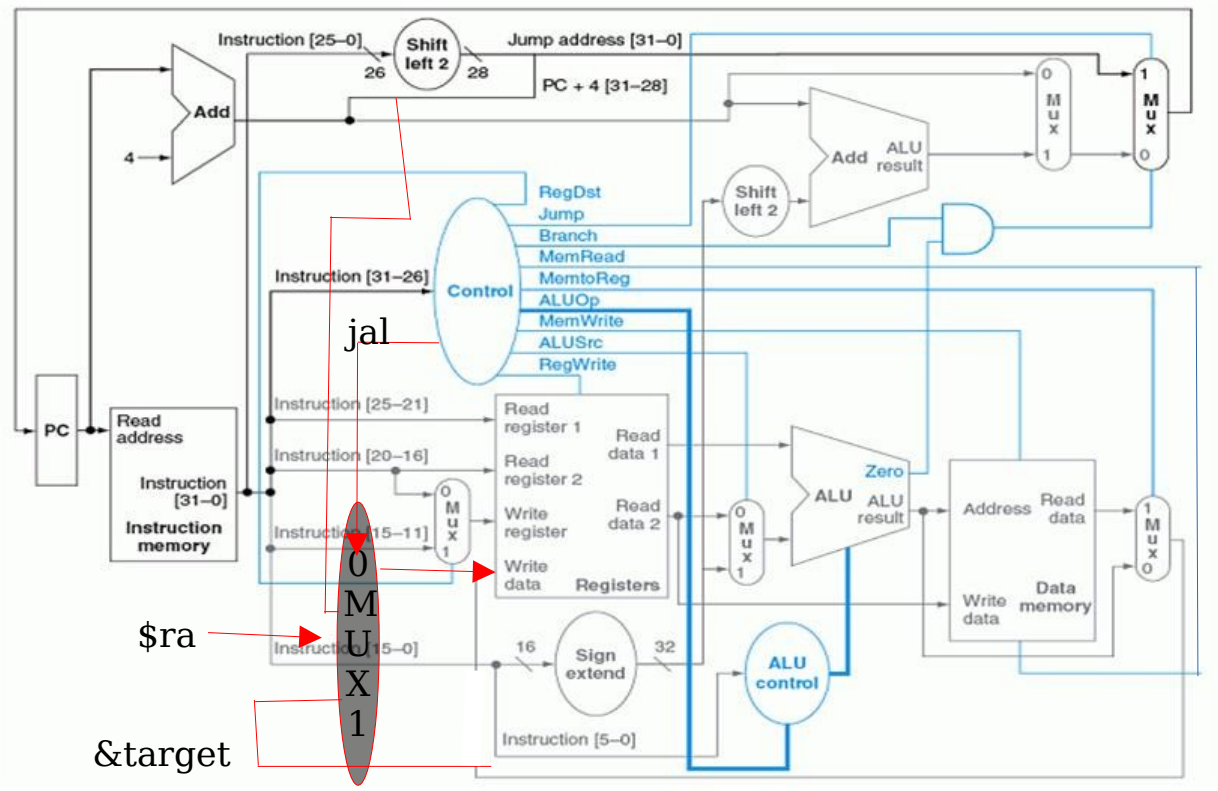

Abbildung 2: Single Cycle CPU mit Steuerwerk

Aufgabe 3.3 Verständnisfragen

3.3.1 Welche der folgenden Aussagen trifft auf einen Ladebefehl zu?

a. MemtoReg muss so gesetzt werden, dass Daten aus dem Speicher an den Registersatz gesendet werden.

-> trifft zu: MemtoReg muss auf 1 gesetzt werden, damit Daten aus dem Speicher an den Registersatz gesendet wird

b. MemtoReg muss so gesetzt werden, dass das richtige Registerziel an den Registersatz gesendet wird.

falsch/nicht

c. Wir kümmern uns nicht um das Setzen von MemtoReg.

-> Wir kümmern uns nicht direkt um das Setzen von MemToReg

3.3.2 Wahr oder falsch?

a. Main Control ist ein kombinatorisches Element mit 6 Eingängen (dem Function-Code) und acht Ausgängen, wobei wie gesehen ALUOp aus zwei Bits besteht.

falsch, dem OPCODE

b. Bei einem „`add`“ Befehl gibt die ALU-Control eine „`0010`“ aus.

wahr

c. Bei einem `add` Befehl wird im Datenblock das Steuersignal "MemWrite" auf "1" gesetzt.

falsch, MemWrite wird auf 0 gesetzt, um ungewollten Speicherzugriff zu verhindern
(Memory wird nicht benutzt)

d. Auf den PC werden prinzipiell 4 Worte addiert, um die nächste Instruktion aus dem Instruktionsspeicher zu laden.

Falsch, 4 Bytes

3.3.3 Betrachten Sie den folgenden Befehl:
Befehl: `and Rd, Rs, Rt`
Interpretation: `Reg[Rd] = Reg[Rs] AND Reg[Rt]`

a. Was sind die Werte der Steuersignale in Abbildung 1, die für den obigen Befehl von der Steuerung generiert werden?

AluSrc: 1, AluOp: 0010, PCSrc: 0, MemWrite: 0, MemToReg: 1, MemRead: 0, RegWrite: 1

b. Welche Ressourcen (Blöcke) führen für diesen Befehl eine nützliche Funktion aus?
PC, Instruction Memory, Register, ADD

c. Welche Ressourcen (Blöcke) erzeugen Ausgaben, die nicht für diesen Befehl verwendet werden? Welche Ressourcen erzeugen für diesen Befehl keine Ausgaben?

DATA MEMORY |=> trotzdem haben alle Blöcke eine Ausgabe

3.3.4 Die einfache Eintakt-MIPS-Implementierung in Abbildung 2 kann nur manche Befehle implementieren. Neue Befehle können zu einer existierenden Befehlssatzarchitektur hinzugefügt werden, aber die Entscheidung, dies zu tun oder zu lassen, hängt unter anderem von den Kosten und der Komplexität ab, die dem Datenpfad und dem Steuerwerk des Prozessors hierdurch aufgeladen werden. Die drei Teilaufgaben dieser Aufgabe beziehen sich auf den folgenden neuen Befehl:

Befehl: `lwr Rt, Rd(Rs)`
Interpretation: `Reg[Rt] = Mem[Reg[Rd]+Reg[Rs]]`

a. Welche existierenden Blöcke (wenn es denn solche gibt) können für diesen Befehl verwendet werden?
PC, Instruction Memory, Register, ALU (für Offset berechnen)

b. Welche neuen funktionalen Blöcke (wenn es denn solche gibt) benötigen wir für diesen Befehl?
Wir brauchen einen extra MUX, damit wir ins Rd-Register schreiben, Statt von ihm zu lesen.

c. Welche neuen Signale vom Steuerwerk (wenn es denn solche gibt) benötigen wir für die Unterstützung dieses Befehls?

lwr-Befehl