

## Übung 6

Abgabe bis **Donnerstag, 13. Juni 10:15** via EPIIC: <https://ep.iic.jku.at>.

### 1 Sequenzielles Steuerwerk (10 Punkte)

Betrachte einen MIPS-Prozessor mit sequenziellem Steuerwerk.

- (a) Erweitere das Steuerwerk und/oder den Datenpfad des Prozessors (Abbildungen 1 und 2), sodass der Befehl `jr` ausgeführt werden kann.
- (b) Bestimme die Länge des kritischen Pfades des erweiterten Prozessors und zeichne ihn in Abbildung 2 ein. Bestimme dazu die Länge aller kritischen Pfade zwischen zwei getakteten Komponenten (z. B. zwischen Registern). Entnimm dabei die Verzögerungen der Komponenten aus Tabelle 1. Die Verzögerung für nicht gelistete Komponenten soll mit  $0\text{ ps}$  angenommen werden.
- (c) Berechne die maximale Taktfrequenz um den Prozessor zu betreiben und vergleiche diese mit der maximalen Taktfrequenz eines Prozessors mit kombinatorischem Steuerwerk<sup>1</sup> (Abbildung 3).

Table 1: Verzögerungen der Prozessorkomponenten

Kombinatorische Komponenten		Sequenzielle Komponenten		
Komponente	$t_{pd}$ [ps]	Komponente	$t_{pcq}$ [ps]	$t_{setup}$ [ps]
Datenspeicher (lesen)	220	Datenspeicher (schreiben)	40	50
Registerbank (lesen)	140	Registerbank (schreiben)	40	45
Instruktionsspeicher (lesen)	250	Register	40	40
ALU	175			
Addierer	125			
2:1-Multiplexer	30			
4:1-Multiplexer	50			
UND/ODER-Gatter	20			

<sup>1</sup>Dazu muss auch für diesen Prozessor der kritische Pfad ermittelt werden.

## 2 Realisierung komplexer Befehle (14 Punkte)

Um häufig wiederkehrende Befehlsfolgen schneller auszuführen, soll der Befehlssatz des MIPS-Prozessors erweitert werden:

- Der I-Type Befehl `addInMem $r0, <offset>($r1)` soll das gespeicherte Wort an der Adresse `<offset>($r1)` (die Berechnung der Adresse erfolgt wie bei den Befehlen `lw` und `sw`) um den Wert des Register `$r0` erhöhen.
- Der R-Typ Befehl `addSub $r0, $r1, $r2, $r3` soll dabei `$r1 + $r2 - $r3` berechnen und das Ergebnis in Register `$r0` abspeichern. Der zusätzliche Operand `$r3` wird in der Maschinensprache anstelle des Shift-Amount codiert.

Erweitere den Prozessor (Datenpfad und/oder Steuerwerk) sodass er die Befehl `addInMem` und `addSub` unterstützt. Achte dabei darauf die minimale Taktlänge möglichst wenig zu verändern (vgl. Aufgabe 1). Bestimme auch wie viele Taktzyklen durch Verwendung dieser neuen Befehle gespart werden um die gewünschte Funktionalität zu erreichen.

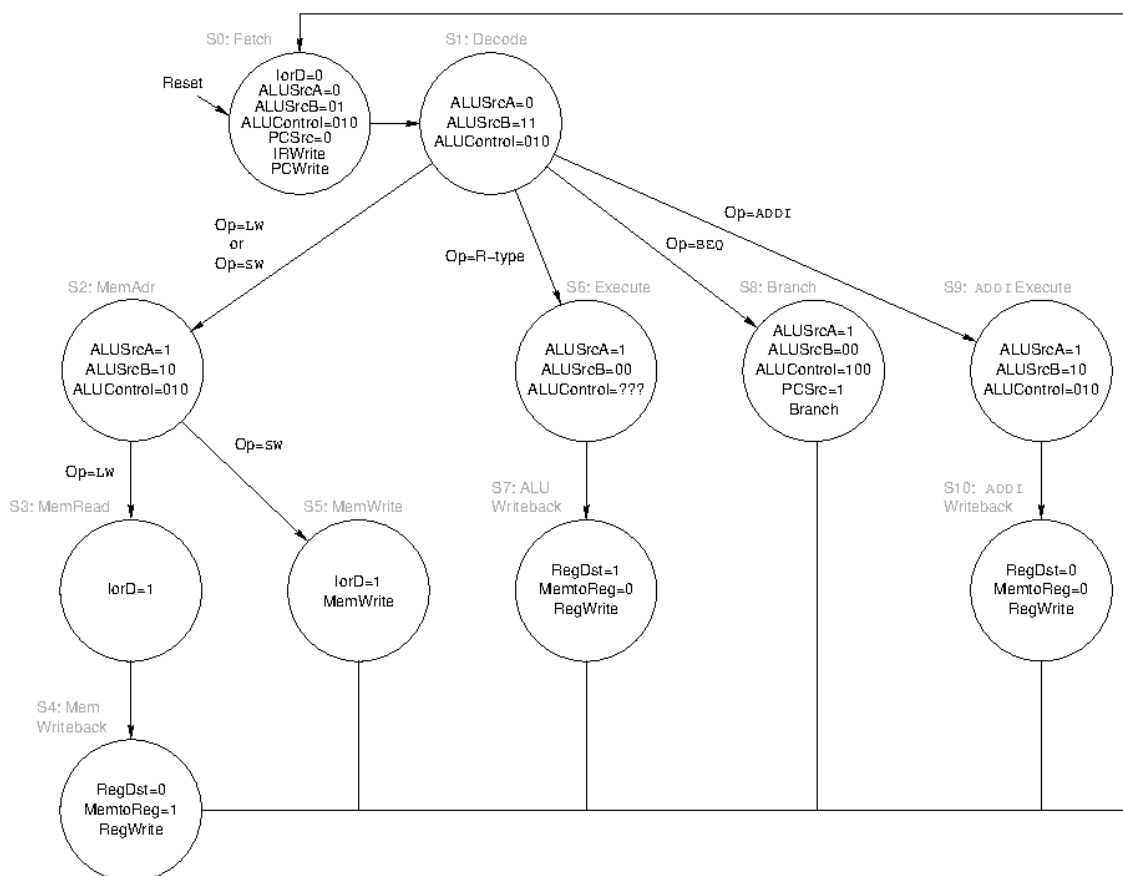


Abbildung 1: Endlicher Automat des sequenziellen Steuerwerks

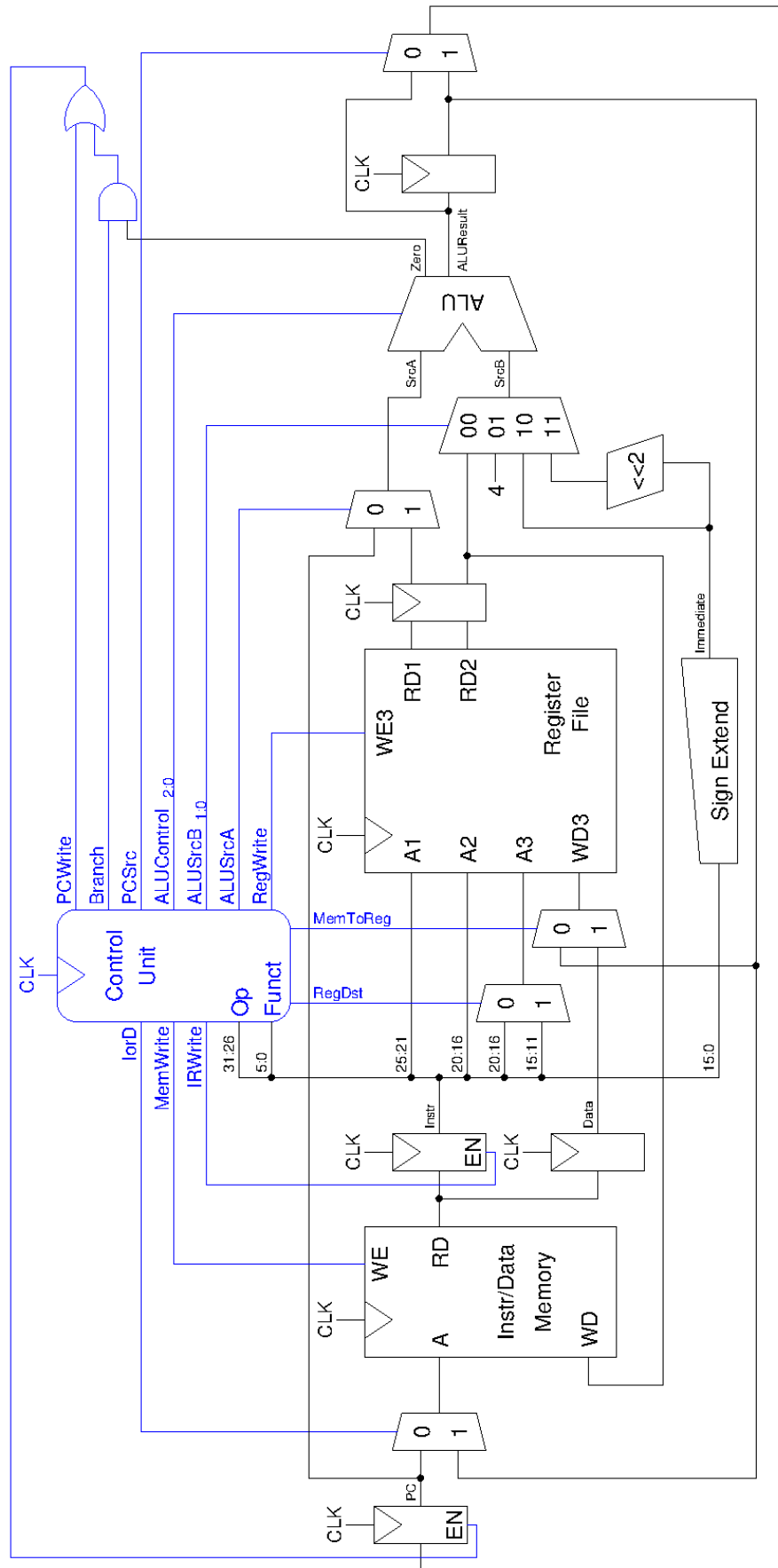


Abbildung 2: MIPS-Prozessor mit sequenziellem Steuerwerk

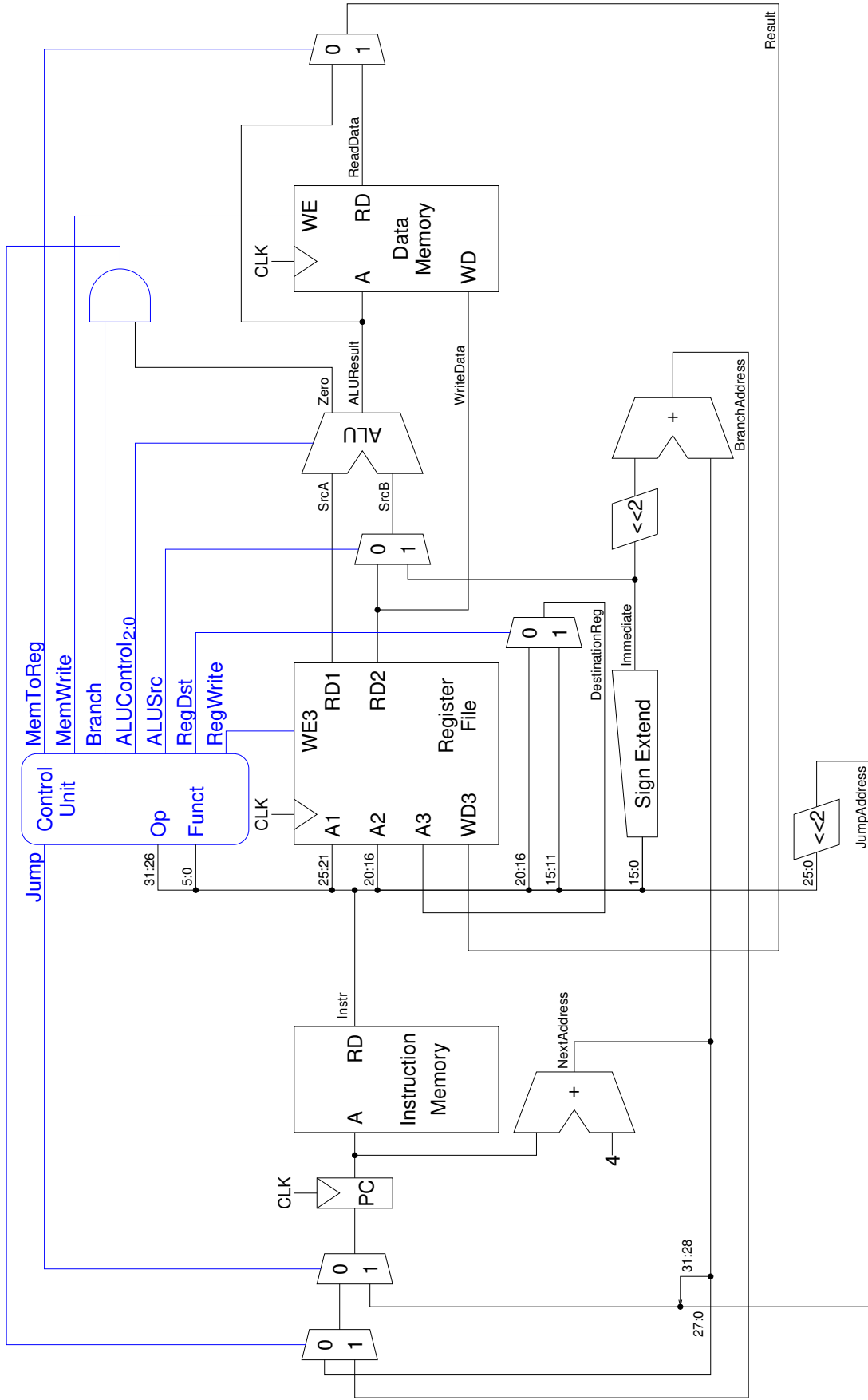


Abbildung 3: MIPS-Prozessor mit kombinatorischem Steuerwerk