
Projekt

Abgabe bis **Donnerstag, 20. Juni 10:15** via EPIIC: <https://ep.iic.jku.at>.

Realisierung eines MIPS-Prozessors (48 Punkte)

Im Rahmen des Projekts soll der in der Vorlesung vorgestellte MIPS-Prozessor erweitert und in VHDL realisiert werden, sodass die Funktion *titlecase* aus Übung 2 ausgeführt werden kann. Das Projekt ist in die folgenden drei Teile unterteilt:

1. Übersetzung von Assembler Code in Maschinensprache

Übersetze die Datei *titlecase.s* mit Hilfe von QtSpim in Maschinensprache und speichere diese in der Datei *imem.dat* ab. Übersetze zusätzlich die globalen Daten in Binärformat (kann ebenfalls mit Hilfe von QtSpim gemacht werden) und speichere diese in der Datei *dmem.dat* ab. In den beiden Dateien beschreiben die ersten 8 Zeichen jeder Zeile ein Datenwort (hexadezimal codiert). Alle weiteren Zeichen in einer Zeile werden ignoriert und können z.B.: für Kommentare genutzt werden. In beiden Dateien ist das zu verwendende Format exemplarisch angeführt. Folgendes ist bei dieser Aufgabe zu beachten:

- Der Speicherbereich für den Programmcode startet bei der Adresse `0x0`, d.h. die erste Instruktion der `main`-Methode soll sich an der Adresse `0x0` befinden. Deshalb müssen die Sprungadressen von J-Typ Befehlen entsprechend angepasst werden.
- Der Speicherbereich für globale Daten startet an der Adresse `0x100`. Der String *str* soll daher an dieser Adresse liegen. Deshalb muss der Pseudobefehl `la` mit `addi` (anstelle von `lui`) ersetzt werden.
- QtSpim weicht bei der Befehlscodierung von bedingten Sprüngen (z.B.: `beq`) vom MIPS Standard ab. Deshalb muss hier bei dem von QtSpim generierten Maschinenwort die 16-Bit Konstante um 1 verringert werden.

2. Konzeptionelle Erweiterung des Prozessors

Der in der Vorlesung vorgestellte MIPS-Prozessor muss erweitert werden, um den die Funktion *titlecase* aus Übung 2 ausführen zu können.

Erweitere den in der Vorlesung vorgestellten MIPS-Prozessor (Abbildung 2), sodass auch die Befehle `jal`, `jr`, `slti`, `bne`, `lb` und `sb` unterstützt werden. Erweitere dazu in Abbildung 2 für jeden dieser Befehle (falls nötig) den Datenpfad und füge weitere Kontrollsignale hinzu.¹ Gib für jeden Befehl auch den Wert aller Kontrollsignale an. Nimm dabei an, dass die ALU aus Abbildung 1 verwendet wird und gib *don't cares* explizit mit *x* an.

¹Der in schwarz gezeichnete Teil in Abbildung 2 visualisiert den Datenpfad, während der Kontrollpfad in blau gezeichnet ist.

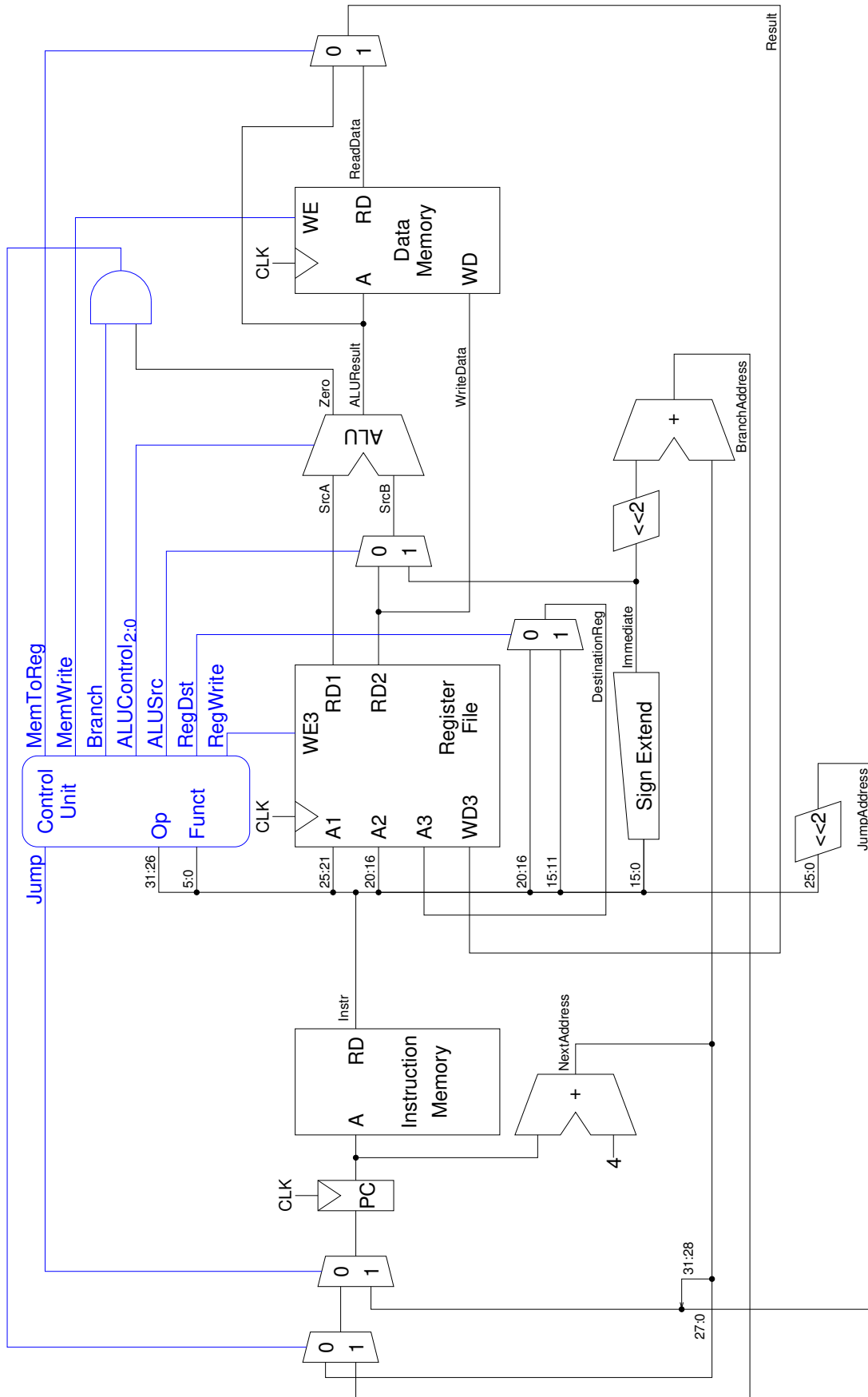


Abbildung 2: MIPS Prozessor