

7. Übung für die Vorlesung Rechnerorganisation

Sommersemester 2019

Abgabe: Donnerstag, 23.5.2019

Aufgabe 1. *Binärdarstellung*

2 P.

Eine Bitfolge hat zunächst keine inhärente Bedeutung. Betrachten Sie das folgende Bit-Muster:

1000 1111 1110 1111 1100 0000 0000 0000

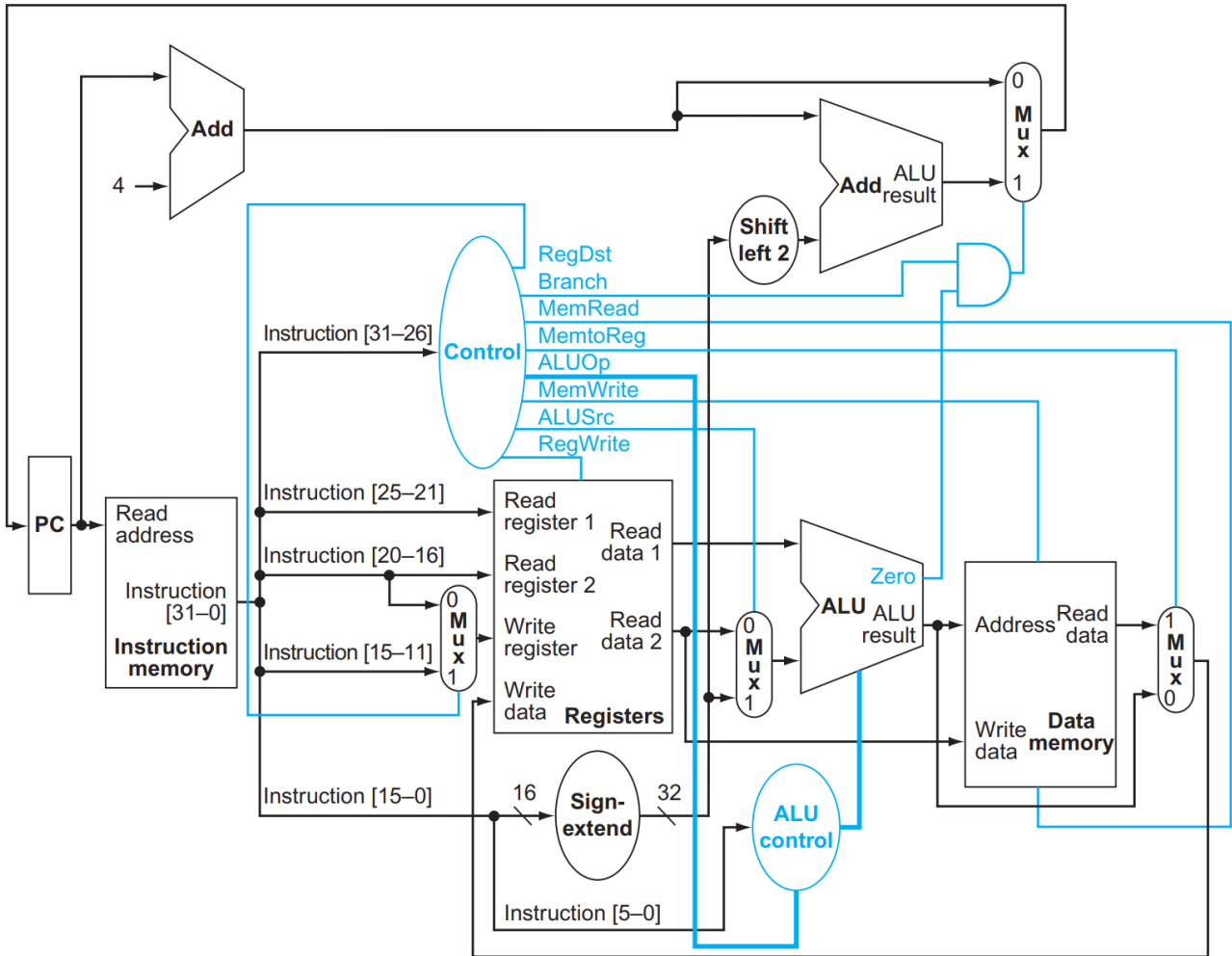
Was repräsentiert es, angenommen, es sei

1. eine vorzeichenlose Ganzzahl?
2. eine Zahl im Zweierkomplement?
3. eine Zahl im IEEE 754 Floating-Point Format?
4. eine MIPS Instruktion?

Aufgabe 2. *Single-Cycle-Datenpfad: addi-Instruktion*

4 P.

Gegeben ist der aus der Vorlesung bekannte Single-Cycle-Datenpfad:



Realisieren Sie die Instruktion `addi` (add immediate). Muss dazu der Datenpfad erweitert werden? Erläutern Sie Ihre Antwort. Falls der Datenpfad ausreicht, tragen Sie die korrekten Steuersignale in Tabelle 1 ein.

Instruktion	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALU-Op1	ALU-Op0
<code>lw</code>	0	1	1	1	1	0	0	0	0
<code>sw</code>	x	1	x	0	0	1	0	0	0
<code>beq</code>	x	0	x	0	0	0	1	0	1
<code>addi</code>									

Tabelle 1: Steuersignale im Single-Cycle Datenpfad

Aufgabe 3. *Stuck-at-Zero Fehler*

6 P.

Betrachten Sie die Darstellung der Kontrolle und des Datenpfades aus Aufgabe 2.

Beschreiben Sie die Effekte eines einzelnen *stuck-at-zero fault* (egal welches Signal eigentlich anliegen sollte, es liegt Null an) auf die Funktionsweise von R-Format Operationen, `lw`, `sw` und `beq`.

Betrachten Sie jeden der folgenden Fehler einzeln:

1. `RegDst = 0`3. `MemtoReg = 0`2. `ALUSrc = 0`4. `Zero = 0`**Aufgabe 4.** *Single-Cycle-Datenpfad: lw ohne Offset*

8 P.

1. Wie könnte der Datenpfad aus Aufgabe 2 vereinfacht werden, wenn bei allen Speicherzugriffsfunktionen auf eine Offsetangabe verzichtet wird? Welche Vorteile hätte dies?
2. Ein Nachteil wäre, dass alle load/store Instruktionen mit einem Offset ungleich Null zu Pseudoinstruktionen würden. Wie würde die Befehlssequenz `lw $t0,104($t1)` ohne Pseudoinstruktionen zu implementieren sein?