

8. Übung für die Vorlesung Rechnerorganisation

Sommersemester 2019

Abgabe: Donnerstag, 6.6.2019

Aufgabe 1. *Verzögerungszeiten im Single-Cycle Datenpfad*

8 P.

Betrachten Sie wiederum den Datenpfad aus Aufgabe 3. Nehmen Sie an, dass die Verzögerungszeiten für die Haupteinheiten folgendermaßen gegeben seien:

- Speicherzugriffe 2 ns (Nanosekunden),
- ALU und Addierer 2 ns,
- Registerzugriffe 1 ns (inkl. PC, auch Schreibzugriffe → Setup-Zeit!).

Die restlichen, kleineren Bauteile, wie beispielsweise MUX, arbeiten verzögerungsfrei.

1. Vervollständigen Sie Tabelle 1, indem Sie die von den verschiedenen Instruktionsklassen benötigten Laufzeiten eintragen.

Instruktion	Instr.-Speicher	lese Reg.	ALU Op.	Datenspeicher	schreibe Reg.	Summe
R-Format						
lw						
sw						
beq						

Tabelle 1: Laufzeiten im Single-Cycle Datenpfad

2. In einem Single-Cycle-Datenpfad muss die Zeitspanne eines Zyklus (wenigstens) so lang sein, dass darin die am längsten währende Instruktion abgearbeitet werden kann. Im obigen Beispiel haben sicherlich alle herausgefunden, dass die `lw` (load word) Instruktion mit 9 ns die längste Zeit in Anspruch nimmt, und damit eine Zyklus-Zeit von 9 ns vorgibt.

Nehmen Sie für die weiteren Aufgabenteile an, dass die Verzögerungszeit des Addierers für die PC-Berechnung (program counter) X ns und die für die Sprungadressen-Berechnung Y ns beträgt.

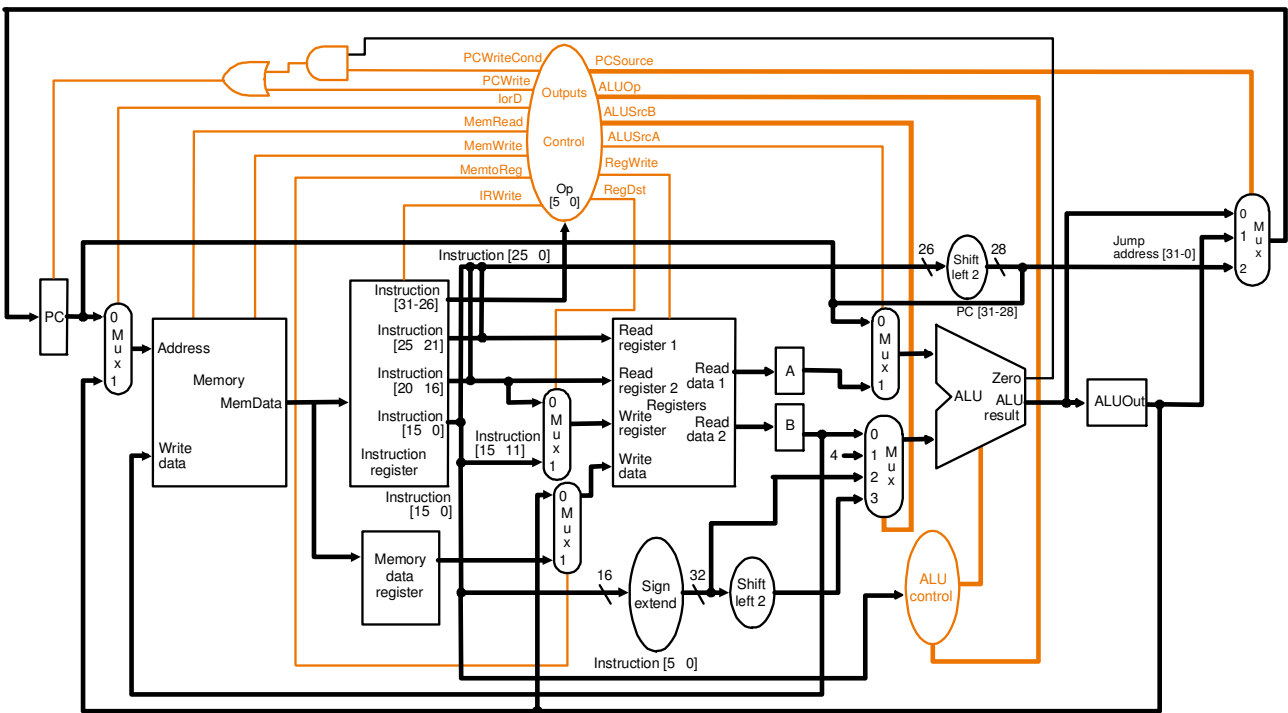
Wie verändert sich die Zyklus-Zeit?

- a) für $X = 3$ und $Y = 3$
- b) für $X = 5$ und $Y = 5$
- c) für $X = 1$ und $Y = 8$

Aufgabe 2. Multi-Cycle-Datenpfad: addi-Instruktion

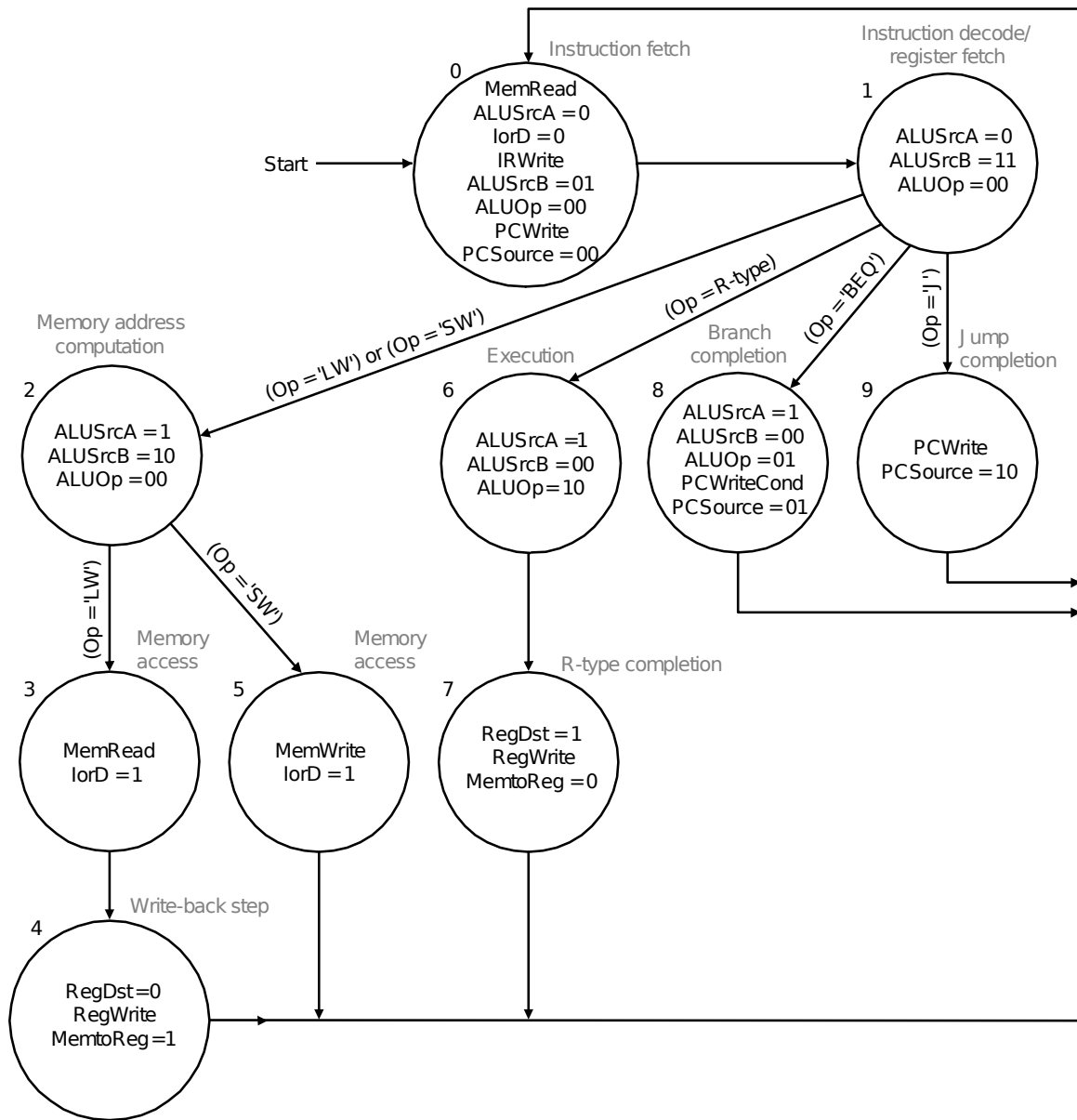
10 P.

Gegeben ist der aus der Vorlesung bekannte Multizyklus-Datenpfad:



Realisieren Sie den Befehl addi (add immediate).

1. Erweitern Sie die graphische Spezifikation der FSM (finite state machine) und die vorhandene Architektur um Datenpfade und Steuerleitungen, falls nötig. Wie muss die Address-select Logic angepasst werden?



2. Erweitern Sie das aus der Vorlesung bekannte Mikro-Programm. Stellen Sie das Mikro-Programm in symbolischer Darstellung und im Binärformat dar.

Aufgabe 3. *Multi-Cycle-Datenpfad: jal-Instruktion*

6 P.

Realisieren Sie den Befehl jal (jump and link).

1. Geben Sie die Befehlsfolge in RTL-Notation an, und passen Sie falls nötig die Multi-Cycle-Architektur aus Aufgabe 1 geeignet an.
2. Wie viele Takte werden benötigt?
3. Welche Steuerleitungen und Datenpfade müssen hinzugefügt werden, um den Befehl in drei Takten abzuarbeiten?

Aufgabe 4. *Multi-Cycle-Datenpfad: jr-Instruktion*

4 P.

Erweitern Sie die graphische Darstellung der FSM zur Steuerung des Multyzyklus-Datenpfades aus Aufgabe 1, so dass der jr -Befehl (jump register) realisiert wird.