



Ausgabe: 22. Januar 2018

Abgaben	{		Theorie	28. Januar 2018
			Praxis	04. Februar 2018
			Rücksprache	05./06. Februar 2018

### **Aufgabe 1: Automaten (5 Punkte)**

Informieren Sie sich (z.B. in [1, ab Seite 273]) über endliche Zustandsautomaten. Erarbeiten Sie insbesondere den Unterschied zwischen Moore- und Mealy-Automaten (z.B. [1, ab Seite 279]).

Beantworten Sie anschließend die folgenden Fragen:

1. (1 Punkt) Was ist ein endlicher Zustandsautomat?
2. (2 Punkte) Wie unterscheiden sich Moore- und Mealy-Automaten voneinander? Nennen Sie für beide Varianten jeweils einen Vorteil!
3. (2 Punkte) In VHDL lassen sich Automaten über ein bis mehrere Prozesse realisieren. Wonach wird bei einer Mehrprozessimplementierung die Logik aufgeteilt? Wie viele sinnvolle Varianten gibt es und welche Vorteile ergeben sich?

### **Aufgabe 2: Zustandsautomat für Mehrzyklenimplementierung (5 Punkte)**

In dieser Aufgabe soll der Zustandsautomat zur Steuerung des einfachen MIPS mit Mehrzyklenimplementierung realisiert werden, wie er in [1, Kapitel 5] beschrieben ist. Der Zustandsgraph des Automaten kann der Abbildung 1 entnommen werden. Implementieren Sie den Zustandsautomat in der Architektur `behavioral` der Datei `mipsCtrlFsm.vhd`.

Testen Sie abschließend Ihr Design mit Hilfe der vorgegebenen Testbench `mipsCtrlFsm_tb`, indem Sie das Kommando `make clean all` in dem Aufgabenordner ausführen.

#### **Hinweis:**

Es lassen sich Aufzählungstypen in VHDL anlegen. Diese sind ideal dazu geeignet, Symbole für Automatenzustände anzulegen. In den Vorgaben befindet sich bereits ein Aufzählungstyp, welchen Sie verwenden sollen. Dieser ist in der Datei `proc_config.txt` beschrieben. Außerdem enthalten die Vorgaben die Datei `mipsISA.txt`, welche Konstanten für die zur Implementierung des Zustandsautomaten relevanten Opcodes enthält.

### **Literatur**

- [1] David A. Patterson and John L. Hennessy. *Rechnerorganisation und -entwurf*. Spektrum Akademischer Verlag, September 2005.

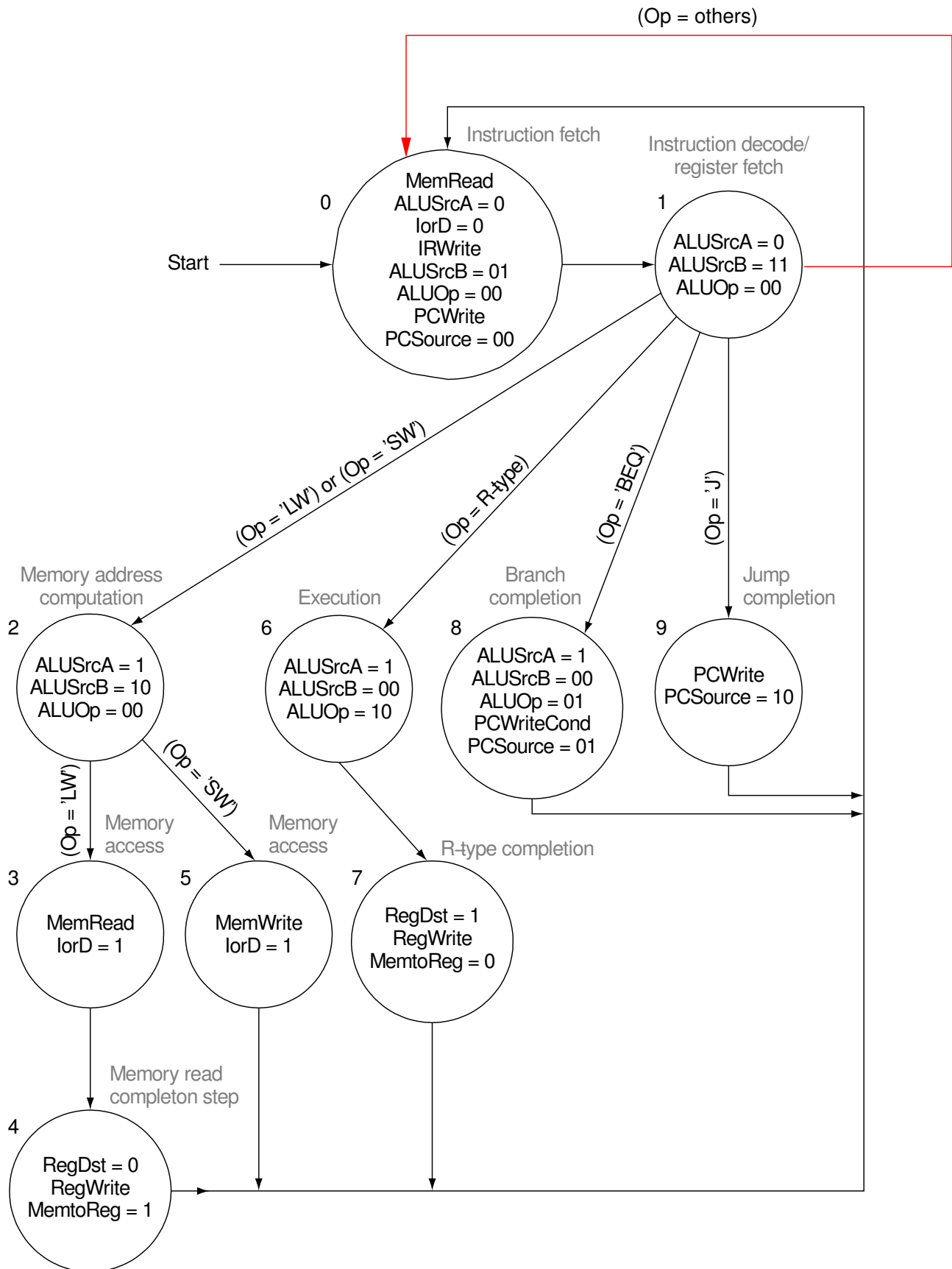


Abbildung 1: Steuerung des Multi-Cycle MIPS-Prozessor mit einem endlichen Automaten