



Ausgabe: 05. Februar 2018

Abgaben	{	 Theorie	entfällt
		 Praxis	11. Februar 2018
		Rücksprache	12./13. Februar 2018

Aufgabe 1: Mehrzyklenimplementierung (10 Punkte)

Realisieren Sie den gesamten MIPS-Prozessor in der Mehrtaktimplementierung. Verbinden Sie dazu alle notwendigen Komponenten in der Architektur `structural`, wie es in Abbildung 1 dargestellt ist. Nutzen Sie die vorgegebene Datei `mipsCpu_mc.vhd`.

Hinweise:

- Es ist möglicherweise einfacher, den gesamten Datenpfad neu zu implementieren, statt die Eintaktimplementierung anzupassen.
- Konsultieren Sie für die Implementierung die Datei `proc_config.txt`. Diese enthält z.B. Konstanten, welche die Breite des PC sowie die Anzahl der Elemente im RAM definieren. Außerdem sind dort die Datentypen der Debug-Ports einzelner Module beschrieben.
- Der RAM arbeitet als Instruktions- und Datenspeicher, da er sowohl das Programm- als auch das Datensegment eines MIPS-Programms enthält.
- Damit die IF-Stufe und die MEM-Stufe innerhalb eines Taktes ausgeführt werden können, muss der Instruktions- und Datenspeicher mit dem invertierten Taktsignal angesteuert werden.
- Beachten Sie, dass der Instruktionsspeicher weniger Einträge hat, als mit dem PC adressiert werden könnten. Üblicherweise wird sich der PC nur in diesem Adressbereich bewegen. Überlegen Sie sich, welcher Teil des PC-Registers zur Adressierung des Speichers verwendet werden muss. Dabei ist außerdem zu beachten, dass der PC eine Byteadresse enthält, während der Instruktionsspeicher wortweise adressiert wird.

1. Implementieren Sie den Datenpfad entsprechend der Abbildung 1. Verbinden Sie außerdem die Debug-Ports von `mipsCtrlFsm`, `regFile` und `ram` mit den entsprechenden Debug-Ports des MIPS, um der Testbench den Zugriff auf den Prozessorstatus zu ermöglichen. Schließen Sie außerdem die Ausgänge der `bin2Char`-Module an den Debug-Port für die 7-Segment-Anzeige an.
2. (4 Punkte) Testen Sie ihre Implementierung mit Hilfe der vorgegebenen Testbench `mipsCpu_mc_mem_tb`, indem Sie im Aufgabenordner das Kommando `make clean vsim/mipsCpu_mc_mem_tb` ausführen. Diese führt auf dem Prozessor ein Programm aus (`memcpy.mif`), das Lade- und Speicherbefehle enthält.
3. (6 Punkte) Testen Sie ihre Implementierung mit Hilfe der vorgegebenen Testbench `mipsCpu_mc_alu_tb`, indem Sie im Aufgabenordner das Kommando `make clean vsim/mipsCpu_`

mc_alu_tb ausführen. Diese führt auf dem Prozessor das Programm clip.mif aus. Das Programm besteht aus Sprung- und ALU-Befehlen.

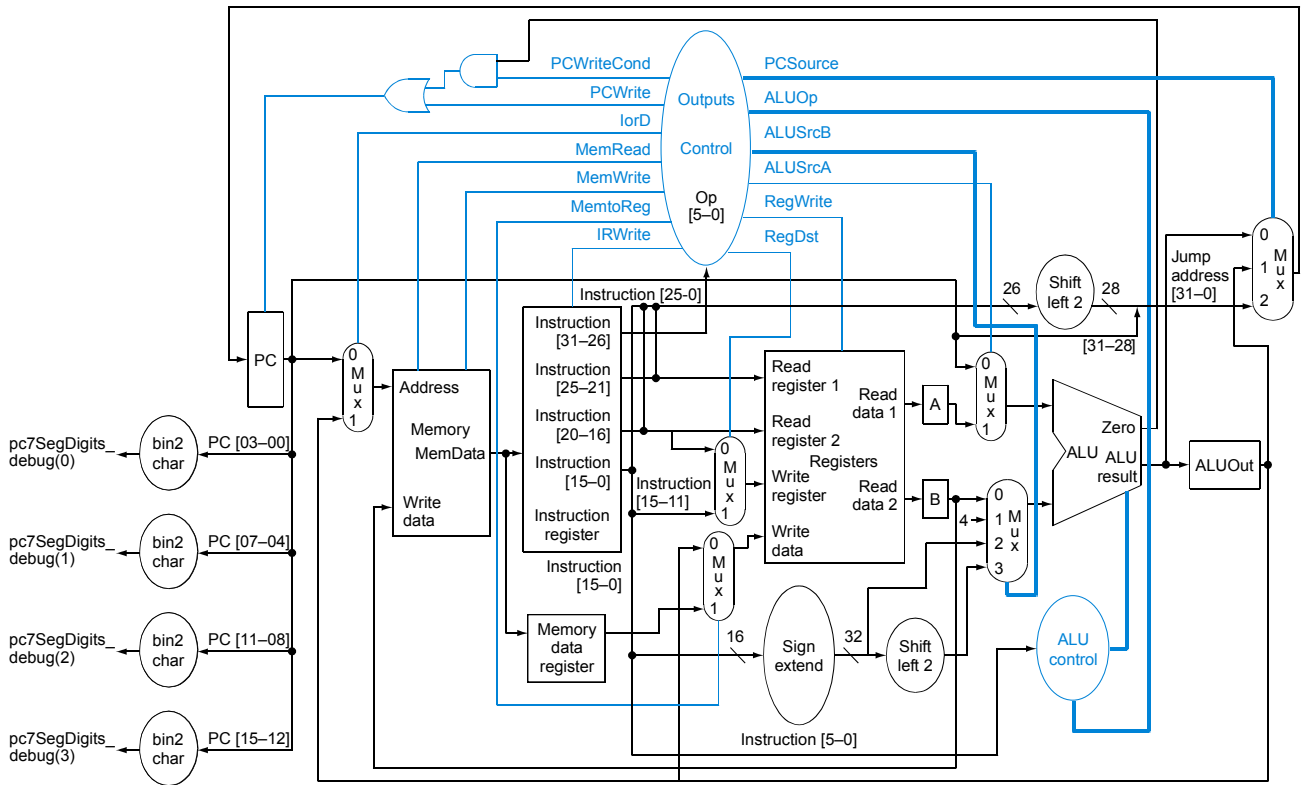


Abbildung 1: Der einfache MIPS mit Steuereinheit, nach [1, S. 266]

Literatur

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