


Alle Aufgaben, die durch ein  gekennzeichnet sind, sind für die eigenständige Vor- bzw. Nachbereitung der Übung und zur Klausurvorbereitung gedacht. Sie werden in der Regel nicht von den Übungsleitern behandelt, können aber ggf. während des Tutoriums selbständig unter Anleitung des Tutors bearbeitet werden, sofern ausreichend Zeit zur Verfügung steht.

4.1 Erstellen einer ALU

In der folgenden Aufgabe soll eine 1-Bit ALU für einen fiktiven MIPS Prozessor erstellt werden. Durch Zusammenschaltung von 32 Stück dieser ALUs ist es möglich eine vollwertige 32-Bit ALU für den MIPS Prozessor zu erstellen. Die ALU soll in der Lage sein folgende Operationen durchzuführen:

- AND
- OR
- ADD
- SUB
- NOR
- NAND

Gehen Sie dabei schrittweise vor.

1. Überlegen Sie sich zunächst wie die ALU aufgebaut sein muss, damit sie die AND und OR Operationen durchführen kann.
2. Erweitern Sie die ALU nun um die ADD und SUB Funktionalität. Überlegen Sie sich dazu zuerst wie man eine 1-Bit Addition im Binären durchführt. Entwerfen sie anschließend einen Halbaddierer und erweitern Sie diesen zu einem Volladdierer. Integrieren Sie dann den Volladdierer in die bisherige ALU.
3. Überlegen Sie nun, wie sich die restliche Funktionalität (NAND, NOR) in die bisherige ALU integrieren lässt.

4.2 Steuersignale der ALU

Überlegen Sie wie die Steuersignale Ainvert, Binvert und Operation gesetzt sein müssen, damit alle Operationen der ALU durchgeführt werden können.

4.3 Vereinfachung der Ansteuerung

Im folgenden soll die Ansteuerung der ALU vereinfacht werden. Es sollen nur noch ein 2-Bit Steuersignal (ALUOp) zur groben Ansteuerung verwendet werden. Dazu ist es nötig eine Logik zur ALU Steuerung (ALUControl) zu entwerfen.

Ist das Steuersignal ALUOp „00“, so soll die ALU unabhängig von dem Function-Code eine Addition durchführen. Die Kombination „01“ veranlasst die ALU unabhängig des Function-Codes eine Subtraktion durchzuführen. Bei der Kombination „10“ bestimmt der „Function-Code“, welche arithmetische

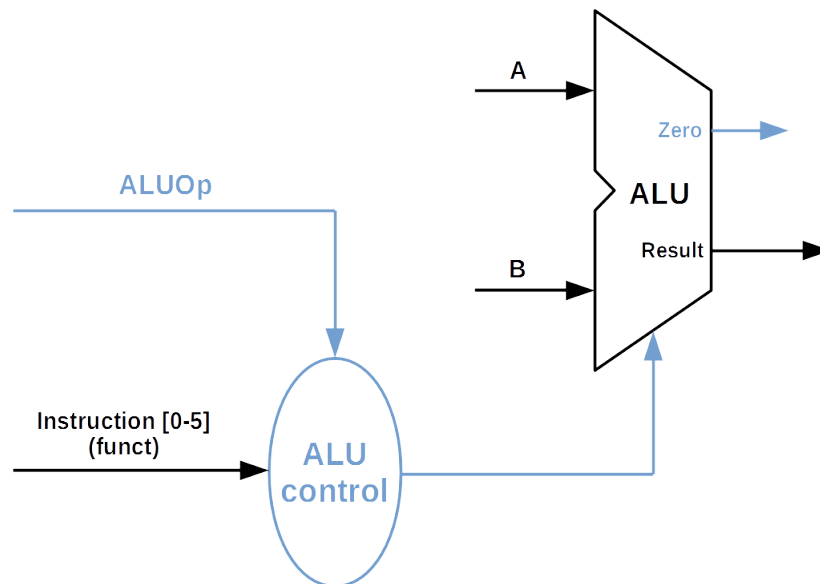


Abbildung 1: Ansteuerung der ALU

Operation (ADD,SUB) ausgeführt wird. Die Kombination „11“ veranlasst die ALU abhängig vom „*Function-Code*“ eine logische Operation auf den Eingängen durchzuführen. Die gesamte Steuerfunktionalität der ALUControl ist der folgenden Tabelle zu entnehmen:

Inputs				Outputs			
AluOp1	AluOp0	F1	F0	Ainv	Binv	Op1	Op0
0	0	X	X	0	0	1	0
0	1	X	X	0	1	1	0
1	0	0	0	0	0	1	0
1	0	0	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	0	1
1	1	1	0	1	1	0	0
1	1	1	1	1	1	0	1

Ermitteln Sie eine Logikfunktion für jeden Ausgang mit Hilfe eines KV-Diagramms. Zeichnen Sie anschließend das entsprechende Schaltbild für die ALU Ansteuerung.