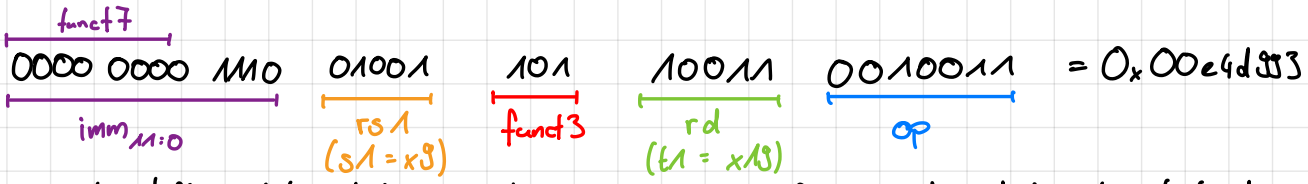
ERA - Übungsblatt 08

1. Weitere siehe ML und Mitschrift WS 23/24

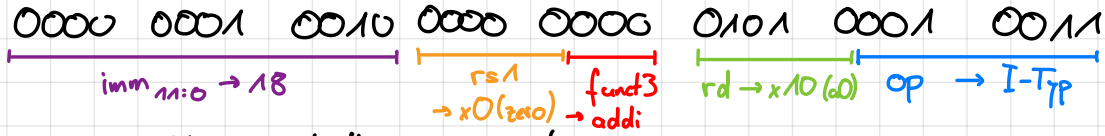
srli s3, s1, 14
Aus der Tabelle kann abgelesen werden, dass srli ein I-Typ ist. Das Layout der einzelnen Felder kann darüber bestimmt werden.



Für alle shifts mit Immediates werden nur 5 Bit für den Immediate benötigt, da nur Shifts von 0 bis 31 möglich sind. In den verbleibenden 12-5 = 7 Bit des Immediate-Felds wird daher funct7 enkodiert.

Disassembly: Das "op"-Feld liegt für alle Befehlstypen an den letzten 7 Bit. Durch den Opcode kann eindeutig der Instruktionstyp festgestellt werden und damit das Layout der anderen Felder

0x01200513:



Die disassemblierte Instruktion lautet also addi a0, zero, 18

2.

Kontrollsignal	Belegung	fehlerhafte Befehle
ALUControl	ADD	• alle R-Typ-Befehle (and, or, sub, ...) außer add, da die ALU natürlich für die Berechnung entsprechend beschaltet werden muss. • alle arithmetisch-logischen I-Typ-Befehle außer addi aus dem selben Grund. Ladebefehle wie lw funktionieren aber, da die ALU nur für Basisadresse aus Register + Offset benötigt wird. • beq, da dafür die ALU eine Subtraktion durchführen muss.
PCSrc	0	• nur bei Befehlen, die evtl. springen (d.h. den nächsten PC auf einen anderen Wert als PC+4 setzen) relevant: beq und jal daher fehlerhaft
ResultSrc	01	• R-Typ und I-Typ, da diese Register beschreiben und daher das ALUResult zurückliefern müssen (außer Ladebefehle). • JAL weil die Rücksprungadresse in ra geschrieben werden muss • andere Befehle sind funktionstüchtig, da RegWrite = 0 ein Beschreiben der Registerbank verhindert.

3, 4. siehe ML und Mitschrift WS 23/24