

TECHNISCHE HOCHSCHULE
MITTELHESSEN

Projektarbeit

SYSTEMROUTINEN FÜR RISC16 PROZESSOR

Arndt Karger

beleitet durch
Prof. Dr.-Ing. Werner Bonath

16. September 2022

Inhaltsverzeichnis

1	Kurzfassung	1
2	Erstellte Routinen	1
2.1	Stackoperationen	1
2.2	Multiplikation via Addition	1
2.3	Linksschieben via Addition	1
2.4	Bitweise Multiplikation	1
3	Zusammenfassung und Fazit	1

1 Kurzfassung

bla

2 Erstellte Routinen

2.1 Stackoperationen

2.2 Multiplikation via Addition

bla

2.3 Linksschieben via Addition

bla

2.4 Bitweise Multiplikation

bla

3 Zusammenfassung und Fazit

bla

Manual für die in der RiSC16 Projektarbeit entworfenen Funktionen:

Registerverwendung:

- r1: zur Wertübergabe, sieht nach UPRO noch gleich aus
- r2: zur Wertübergabe, sieht nach UPRO noch gleich aus
- r3: Ergebnisregister
- r4: Zählregister innerhalb von Funktionen
- r5: don't-care
- r6: Rücksprungadressen
- r7: Stackpointer

Funktionsaufruf:

- Definiere r5, r6 für Programmaufrufe
1. hole Adresse von call in r1:
 movi r5, call
 2. springe zu Adresse in r5, speichere Rücksprungadresse in r6:
 jalr r6, r5
 3. spring wieder zurück: (Hier ist die Rücksprungadresse (r5) unwichtig)
 jalr r5, r6

Stack:

- Definiere SP als r7
- push:
 addi r7, r7, -1 #SP erniedrigen
 sw r1, r7, 0 #r1 weg pushen
 - pop:
 lw r3, r7, 0 #in r3 poppen
 addi r7, r7, 1 #SP erhöhen, damit SP immer auf letztes
 #Ereignis zeigt

shift_l:

- shiftweite (n) in r1
- zu shiftendes Wort (a) in r2
- Ergebnis wird in r3 zurück gegeben

MULv2: (MUL via add)

- Multiplikator in r1
- Multiplikant in r2
- Ergebnis wird in r3 zurück gegeben

MULv3: (bitweise MUL)

- Multiplikator in r1
- Multiplikant in r2
- Ergebnis wird in r3 zurück gegeben

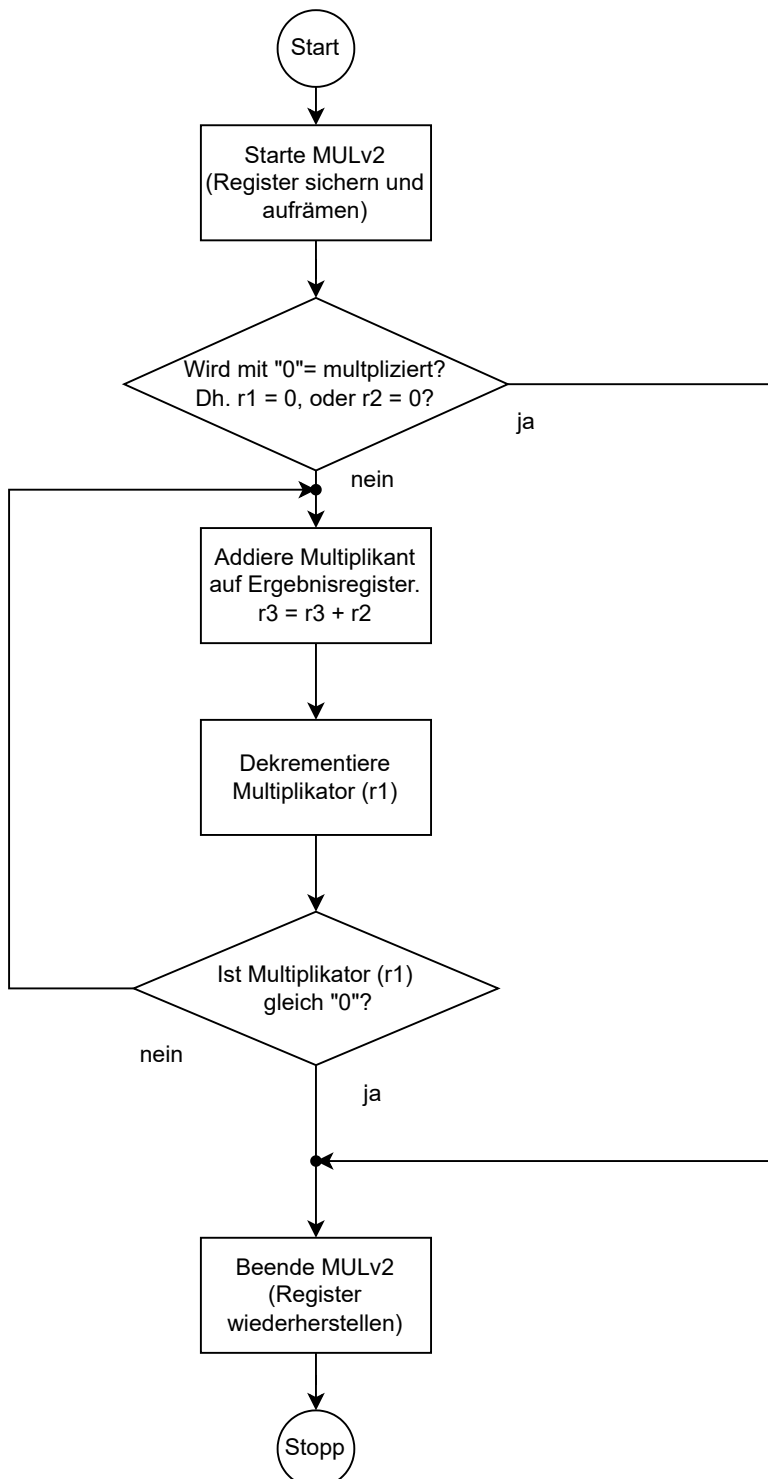
Multiplikation via Addition: (MULv2)

Funktionalität:

Multipliziere zwei vorzeichenlose Zahlen

Verwendung:

- $r3 = r1 * r2$
- Multiplikator und Multiplikant vor Programmaufruf in r1 und r2 laden.
- Diese stehen auch nach Programmablauf unverändert dort.



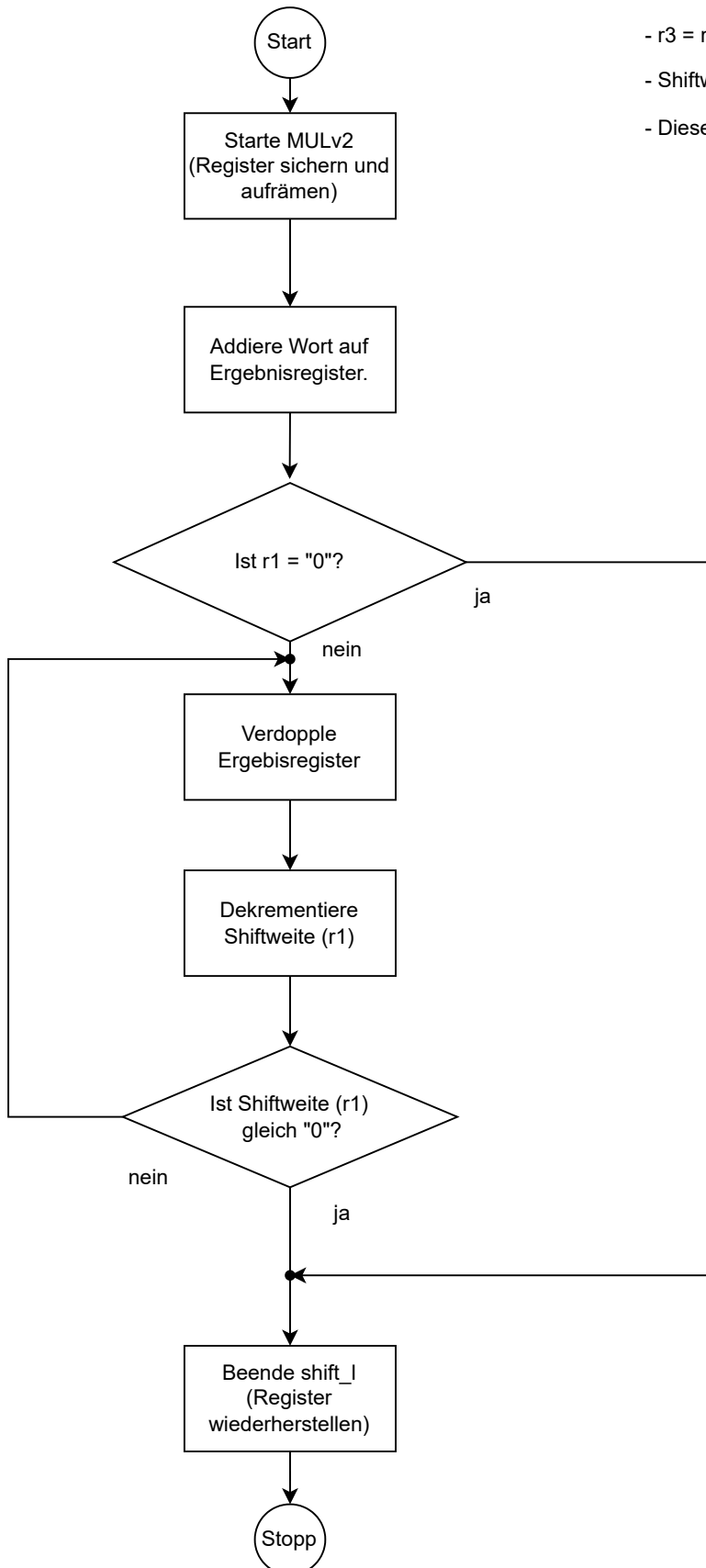
Linksschieben via Addition: (shift_l)

Funktionalität:

Schiebe ein Wort (r2) um eine Shiftweite (r1) nach links

Verwendung:

- $r3 = r2 \ll r1$ und $r1, r2 \neq 0$
- Shiftweite und Wort vor Programmaufruf in r1 und r2 laden.
- Diese stehen auch nach Programmablauf unverändert dort.



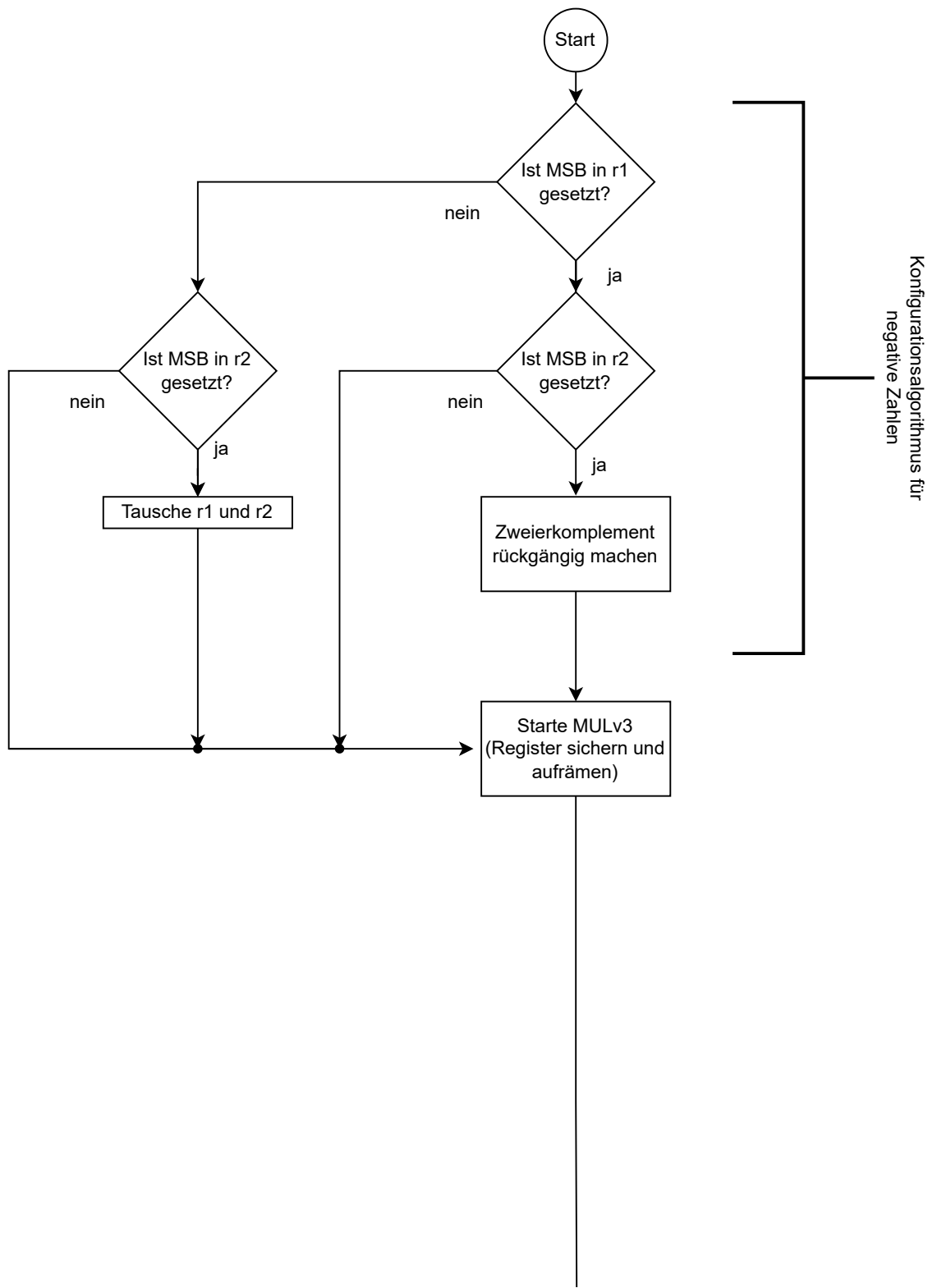
Bitweise Multiplizieren: (MULv3)

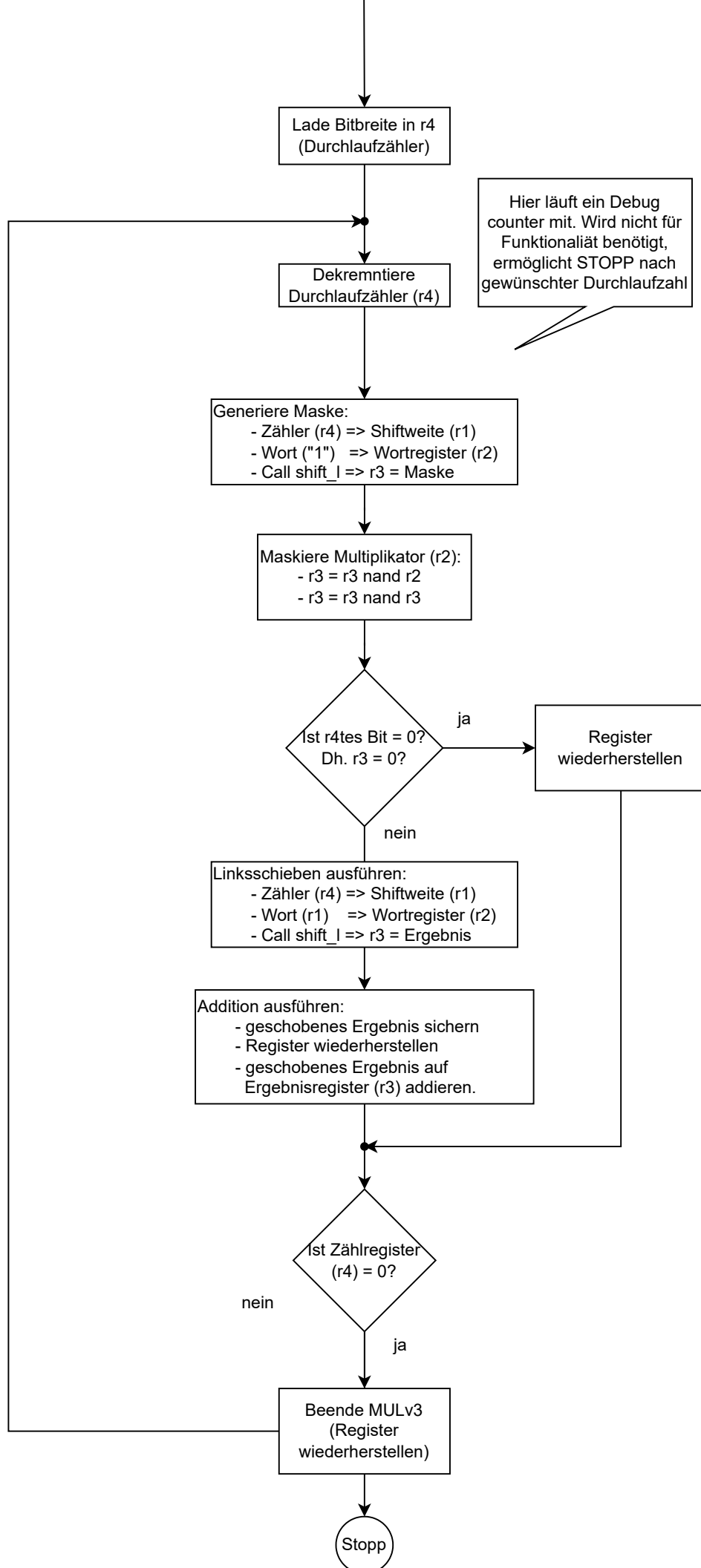
Funktionalität:

Multipliziere zwei vorzeichenbehaftete Acht-Bit-Zahlen

Verwendung:

- $r3 = r1 * r2$
- Multiplikator und Multiplikant vor Programmaufruf in r1 und r2 laden.
- Diese stehen auch nach Programmablauf unverändert dort.





Literatur

- [Ins04] Texas Instruments. *[...] tl084, [...] JFET-INPUT operational amplifiers*. 2004.