

CT-H Projekt 3

2.1.1 Bauteile L1 und L2

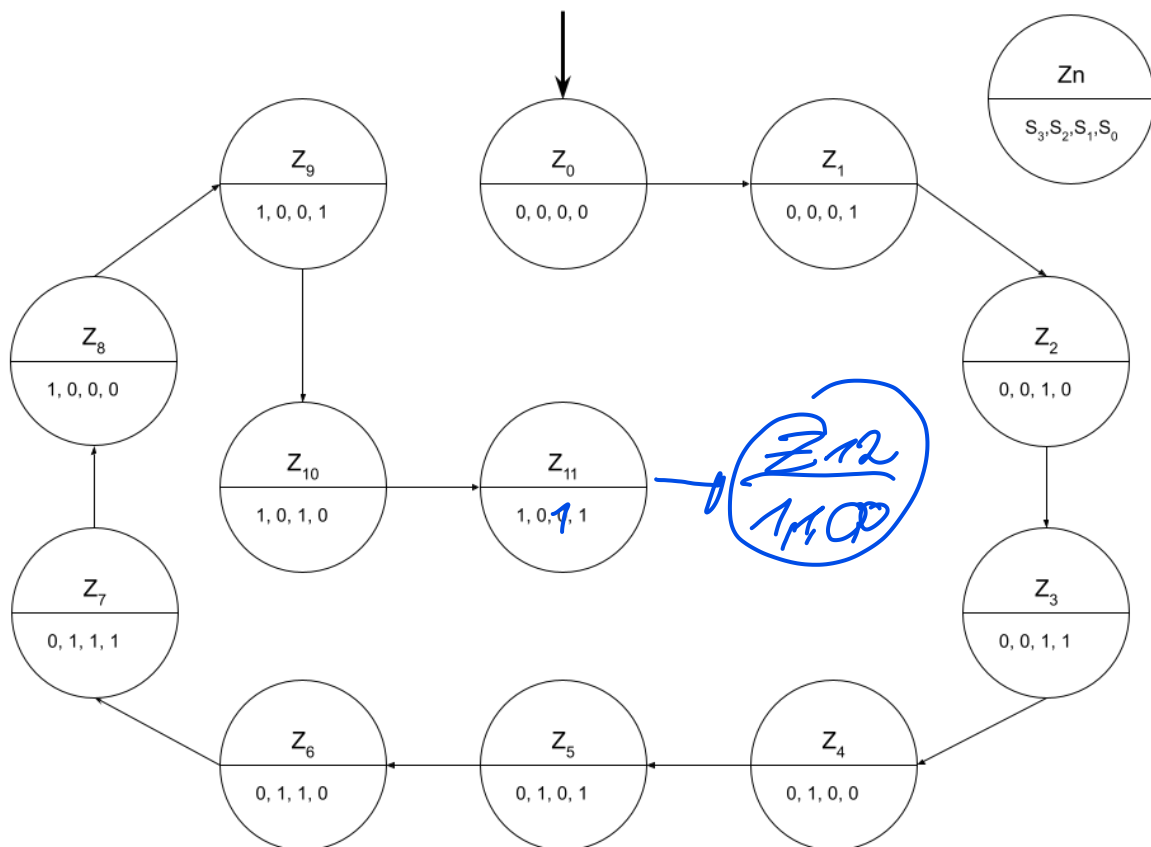
L1: NAND, weil es immer eine 1 ausgibt wenn die Eingänge nicht alle an sind (mindestens 1 aus ist)

L2: AND: es müssen beide 1 sein, sodass am Ausgang eine 1 ausgegeben wird



2.1.2 Zustandsdiagramm von SW und SN1

(S_3, S_2, S_1, S_0 sind die Speicher, die für dieses Schaltwerk nötig sind)



2.1.3 Anzahl benötigter Leitungen + Zustandsfolgetabelle

Man braucht 4 Leitungen, da man 4 Speicher benötigt. ($\log_2(12) = 3,58 \rightarrow 4$)

	t				t + 1				
Zustand	S ₃	S ₂	S ₁	S ₀	S ₃	S ₂	S ₁	S ₀	Folgezustand
z ₀	0	0	0	0	0	0	0	1	z ₁
z ₁	0	0	0	1	0	0	1	0	z ₂
z ₂	0	0	1	0	0	0	1	1	z ₃
z ₃	0	0	1	1	0	1	0	0	z ₄
z ₄	0	1	0	0	0	1	0	1	z ₅
z ₅	0	1	0	1	0	1	1	0	z ₆
z ₆	0	1	1	0	0	1	1	1	z ₇
z ₇	0	1	1	1	1	0	0	0	z ₈
z ₈	1	0	0	0	1	0	0	1	z ₉
z ₉	1	0	0	1	1	0	1	0	z ₁₀
z ₁₀	1	0	1	0	1	0	1	1	z ₁₁
z ₁₁	1	0	1	1	1	1	0	0	z ₁₂
z ₁₂	1	1	0	0					

2.1.4 Wahrheitstabelle von SN1

Zustand	S ₃	S ₂	S ₁	S ₀	A	L
z ₀	0	0	0	0	0	0
z ₁	0	0	0	1	0	0
z ₂	0	0	1	0	1	0
z ₃	0	0	1	1	1	0
z ₄	0	1	0	0	0	0
z ₅	0	1	0	1	0	0
z ₆	0	1	1	0	1	0
z ₇	0	1	1	1	0	0
z ₈	1	0	0	0	0	0
z ₉	1	0	0	1	1	0
z ₁₀	1	0	1	0	1	0
z ₁₁	1	0	1	1	1	0
z ₁₂	1	1	0	0	0	1

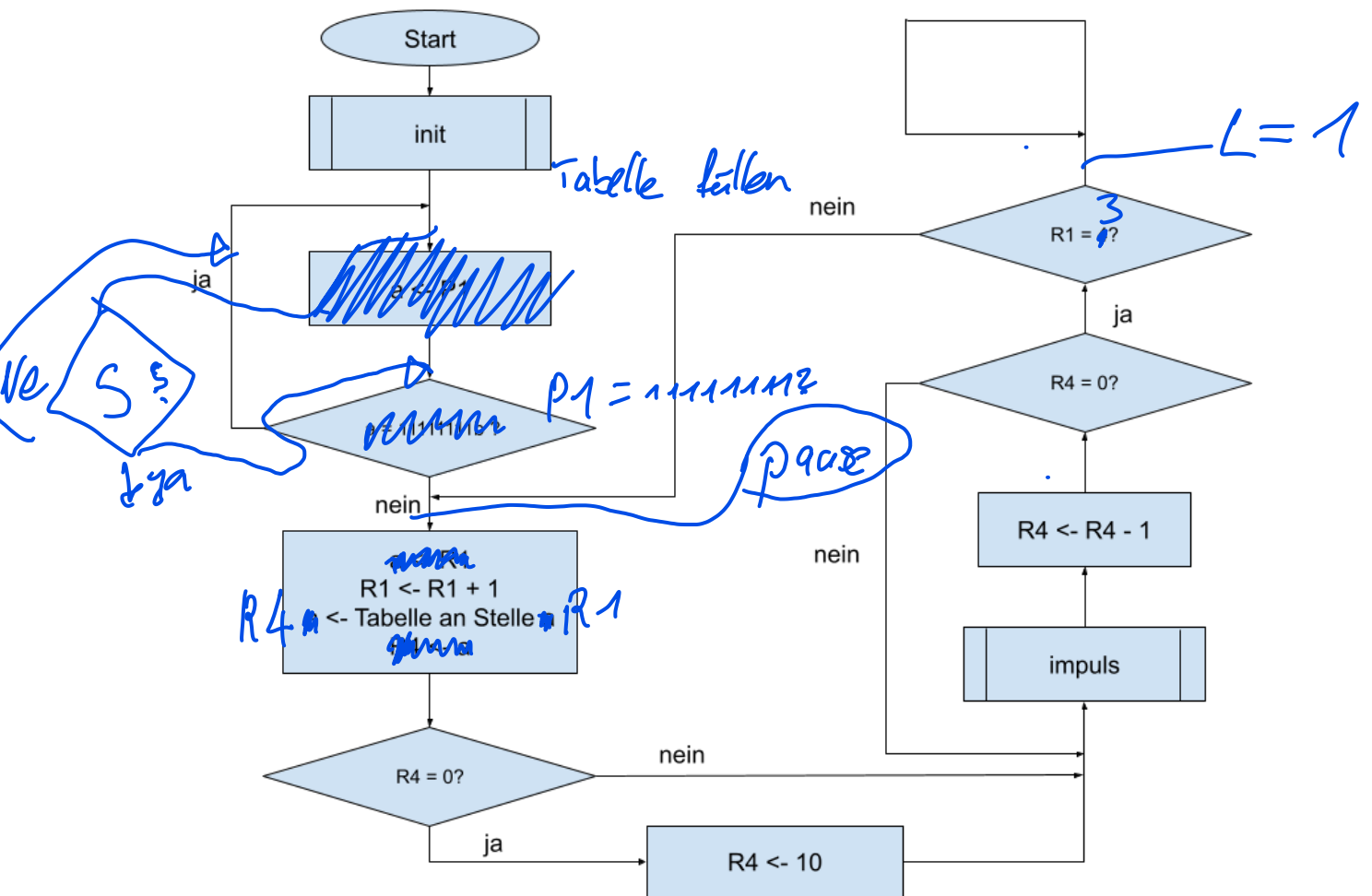
2.2.1 Lösungsansatz für Unterprogramm Impuls (Zählschleife)

Ich würde es mit Hilfe von verschachtelten Zählschleifen realisieren.

Bei Assembler ist dies leichter zu berechnen, da Befehle eine bestimmte Zeit zur Ausführung benötigen. In C ist dies komplizierter, da es sich bei C um eine Hochsprache handelt und nicht ganz so hardwarenah wie Assembler ist. Das heißt in C muss man die Verzögerung, wenn man Zählschleifen verwendet, mit probieren finden.

→ Daher Interrupt nötig bei C

2.2.2 PAP / Struktogramm für Hauptprogramm



2.2.3 Timer-Interrupt Initialisierung, ISR

init_delay_50_timer:

```

setb ea; globale interrupt freigabe
setb et0; ueblerauf timer0 freigabe
mov tmod, #1d ; modus 1 -> 16 bit timer (ohne Nachladen)
mov th0, #3Ch; 3C in th0
mov tl0, #0b0h; B0 in tl0    -> 3CB0 -> 50 ms
setb tr0; startet timer
ret

```

; ISR

org 0Bh

clr TR0; stoppt timer

;timer neu laden

cpl Ausgang

mov th0, #3Ch; 3C in th0

mov tl0, #0B0h; B0 in tl0 -> 3CB0 -> 50 ms

setb tr0; startet timer

reti

2.2.4 Zählschleife/Timer

	Zählschleife	Timer
Prozessorbelastung	sehr hoch, da der mc mit zählen beschäftigt ist	überhaupt nicht, kann also noch andere Befehle verarbeiten
Genauigkeit	durch Rechenfehler können sich schnell ungenauigkeiten einschleichen	wenn man es richtig berechnet, sehr genau
Sonstiges	mc ist beschäftigt	mc kann anderes noch machen

