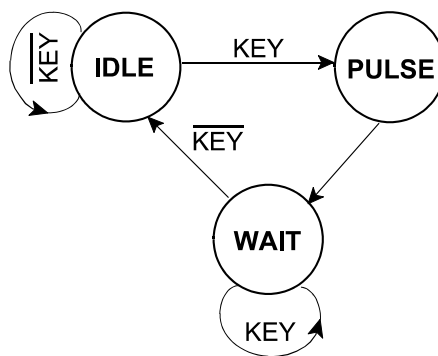


Zustandsautomaten mit OHE-Codierung entwerfen

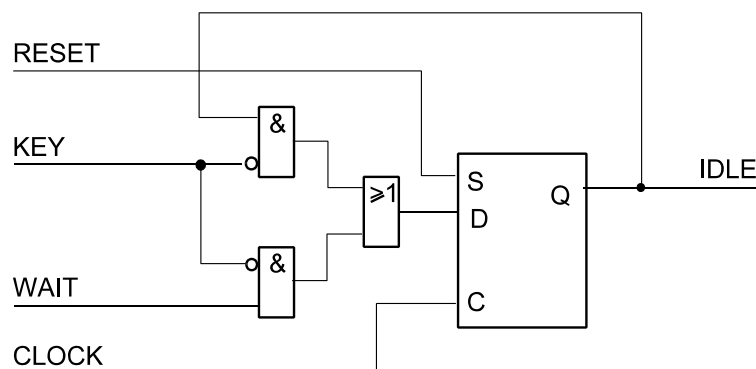
28. 1. 10

Beispiel: der Single-Shot-Generator

Diese State Machine soll auf eine Tastenbetätigung (KEY) hin einen einzigen Impuls (PULSE) abgeben. Das Problem: Die Tastenbetätigung dauert viel länger als eine einzelne Taktperiode. Deshalb ist ein Wartezustand (WAIT) einzuführen, in dem auf das Loslassen der Taste gewartet wird.



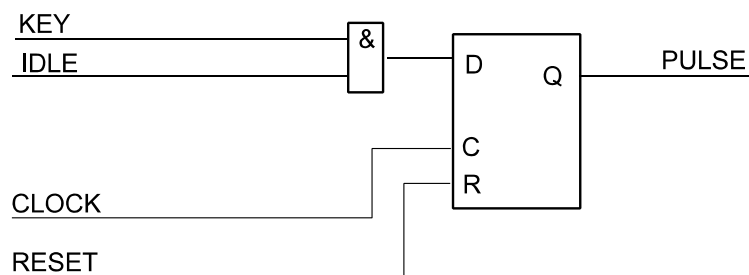
Implementierung mit D-Flipflops:



Der IDLE-Zustand wird eingeleitet, wenn im WAIT-Zustand die Taste losgelassen wird.

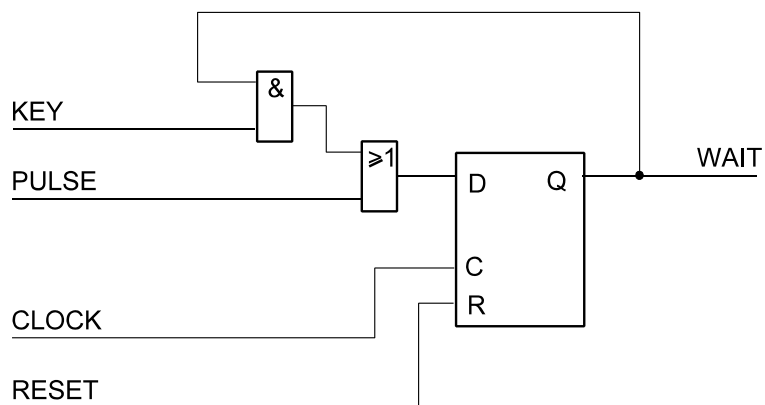
Er wird gehalten, solange die Taste nicht betätigt (= losgelassen) ist. Bei betätigter Taste wird er verlassen.

$$IDLE_D = WAIT \cdot \overline{KEY} \vee IDLE \cdot \overline{KEY}$$



Der PULSE-Zustand wird eingeleitet, wenn im IDLE-Zustand die Taste betätigt wird. Er wird mit dem nächsten Takt wieder verlassen.

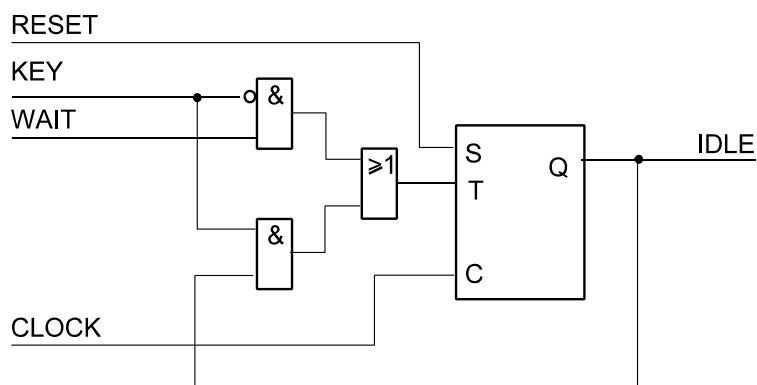
$$\text{PULSE}_D = \text{IDLE} \cdot \text{KEY}$$



Der WAIT-Zustand wird eingeleitet, wenn der PULSE-Zustand aktiv ist. Er wird gehalten, solange die Taste betätigt ist. Bei losgelassener Taste wird er verlassen.

$$\text{WAIT}_D = \text{PULSE} \vee \text{WAIT} \cdot \text{KEY}$$

Implementierung mit T-Flipflops:

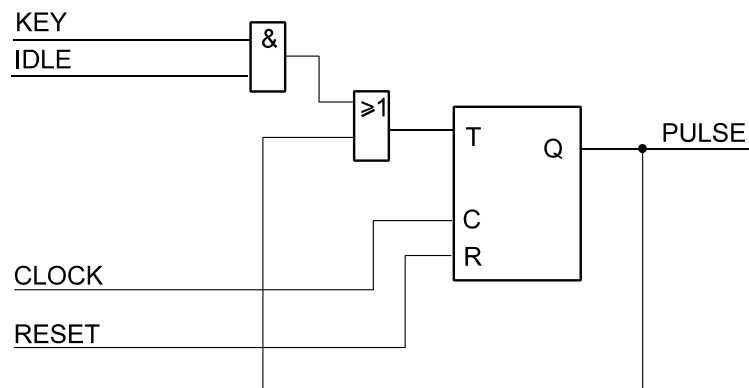


Der IDLE-Zustand wird eingeleitet, wenn im WAIT-Zustand die Taste losgelassen wird.

Er wird verlassen, wenn er aktiv ist und die Taste betätigt wird.

In beiden Fällen ist eine Änderung (T-Bedingung) zu veranlassen.

$$\text{IDLE}_T = \text{WAIT} \cdot \overline{\text{KEY}} \vee \text{IDLE} \cdot \text{KEY}$$

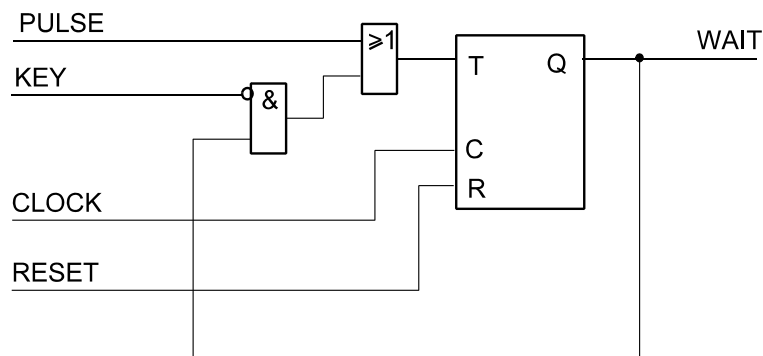


Der PULSE-Zustand wird eingeleitet, wenn im IDLE-Zustand die Taste betätigt wird.

Er wird verlassen, wenn er aktiv ist (= mit dem nächsten Takt).

In beiden Fällen ist eine Änderung (T-Bedingung) zu veranlassen.

$$\text{PULSE}_T = \text{IDLE} \cdot \text{KEY} \vee \text{PULSE}$$

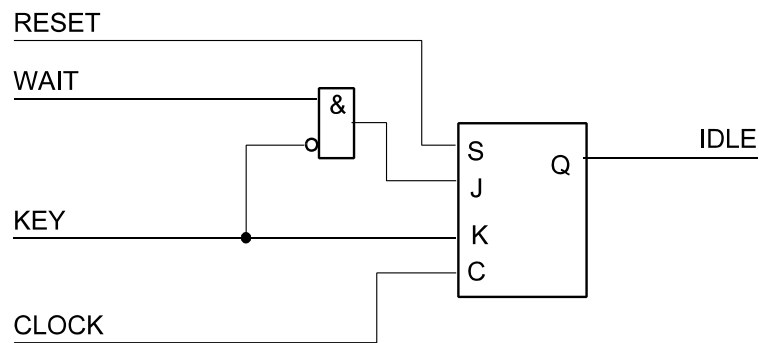


Der WAIT-Zustand wird eingeleitet, wenn der PULSE-Zustand aktiv ist.

Er wird verlassen, wenn die Taste nicht mehr betätigt (= losgelassen) ist.

In beiden Fällen ist eine Änderung (T-Bedingung) zu veranlassen.

$$\text{WAIT}_T = \text{PULSE} \vee \text{WAIT} \cdot \overline{\text{KEY}}$$

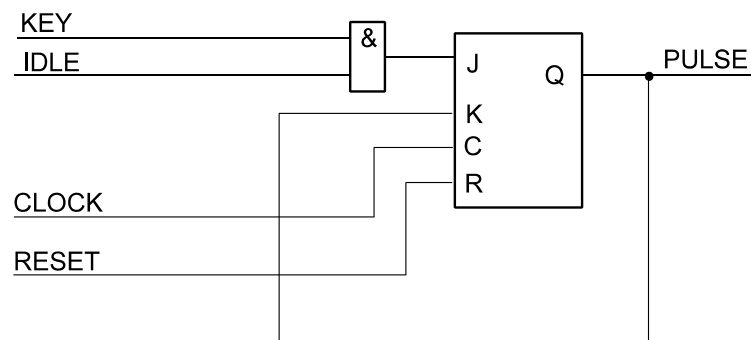
Implementierung mit JK-Flipflops (RS-Funktion):

Der IDLE-Zustand wird eingeleitet, wenn im WAIT-Zustand die Taste losgelassen wird.

Er wird verlassen, wenn die Taste betätigt wird.

$$IDLE_J = WAIT \cdot \overline{KEY}$$

$$IDLE_K = KEY$$

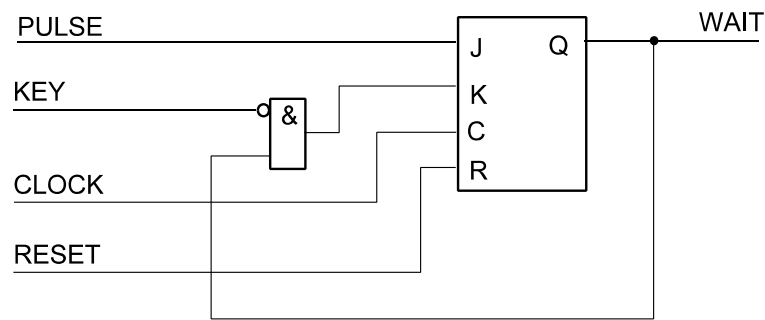


Der PULSE-Zustand wird eingeleitet, wenn im IDLE-Zustand die Taste betätigt wird.

Er wird verlassen, wenn er aktiv ist (= mit dem nächsten Takt).

$$PULSE_J = IDLE \cdot KEY$$

$$PULSE_K = PULSE$$



Der WAIT-Zustand wird eingeleitet, wenn der PULSE-Zustand aktiv ist.

Er wird verlassen, wenn die Taste nicht mehr betätigt (= losgelassen) ist.

$$\text{WAIT}_J = \text{PULSE}$$

$$\text{WAIT}_K = \text{WAIT} \cdot \overline{\text{KEY}}$$

Aufwandsvergleich durch Auszählen der Gatter:

- mit D-Flipflops: 6 Gatter,
- mit T-Flipflops: 7 Gatter,
- mit JK-Flipflops: 3 Gatter.

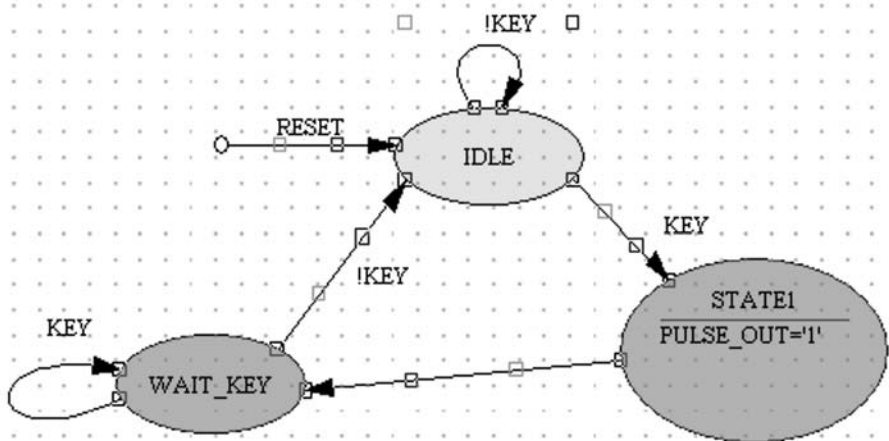
Ersichtlicherweise sind bei den T- und JK-Flipflops die UND-Verknüpfungen der Übergangsbedingungen des jeweils vorhergehenden Zustandes gleich denen der Setzbedingungen des jeweiligen Folgezustandes. Sie müssen also in der Schaltung nur einmal vorgesehen werden (Aufwandsersparnis; bei den T-Flipflops 2 Gatter, bei den JK-Flipflops 1 Gatter). Andererseits sind die T- und JK-Flipflops intern aufwendiger als das D-Flipflop (sie bestehen heutzutage meist aus D-Flipflops mit vorgeschalteten Rückführungsnetzwerken).

Hinweis:

Die hier dargestellten State Machines setzen vollsynchroner Signale voraus (alle Signale auf den gemeinsamen Takt bezogen). Ggf. sind Schaltmittel zur Eintaktierung (Synchronisation), Entprellung usw. vorzuordnen.

Single Shot Generator mit StateCAD

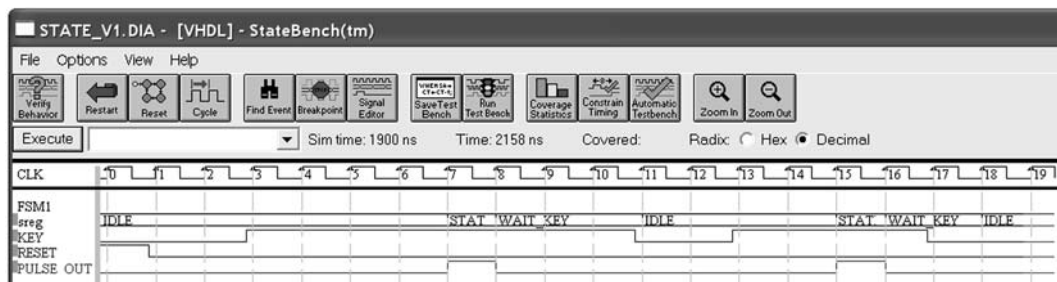
Eingabe des Zustandsdiagramms:



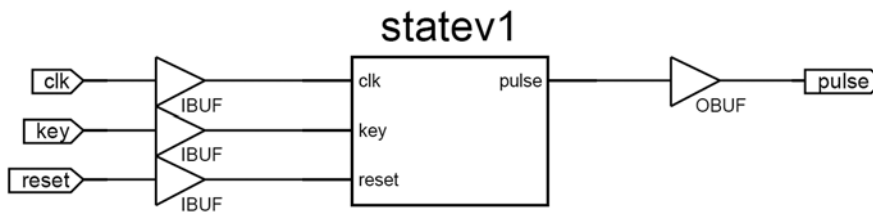
Konfigurationsmenü. Es wurde One Hot Encoding ausgewählt.



Simulation (StateBench):



Das Schaltsymbol:



Die Schaltgleichungen:

```

FDCPE_IDLE: FDCPE port map (IDLE,IDLE_D,CLK,'0',RESET);
  IDLE_D <= ((NOT KEY AND IDLE.LFBK)
  OR (NOT KEY AND WAIT_KEY.LFBK));
FDCPE_PULSE_OUT: FDCPE port map (PULSE_OUT,PULSE_OUT_D,CLK,RESET,'0');
  PULSE_OUT_D <= (KEY AND IDLE.LFBK);
FDCPE_WAIT_KEY: FDCPE port map (WAIT_KEY,WAIT_KEY_D,CLK,RESET,'0');
  WAIT_KEY_D <= ((STATE1.LFBK)
  OR (KEY AND WAIT_KEY.LFBK));

```

Dese Gleichungen entsprechen denen der D-Flipflop-Implementierung auf S. 1 und 2.