

1	Beispiel 1: Entprellen eines Tasters	1
2	Beispiel 2: Drehgeber	3
2.1	Aufbau des Gebers	3
2.2	Folgeautomat	4
2.3	Auswerteautomat	5

1 Beispiel 1: Entprellen eines Tasters

So gut wie alle mechanischen Taster oder Schalter können beim Umschalten prellen. Darunter versteht man die Ausgabe mehrerer Signalwechsel anstelle des eigentlich gewünschten einzelnen Signalwechsels. Die Ursache ist meist eine mechanische Schwingung einer Schaltzunge. Abbildung 1 zeigt den möglichen Zeitverlauf eines Ausgangssignals T eines Tasters auf einen Tastendruck D .

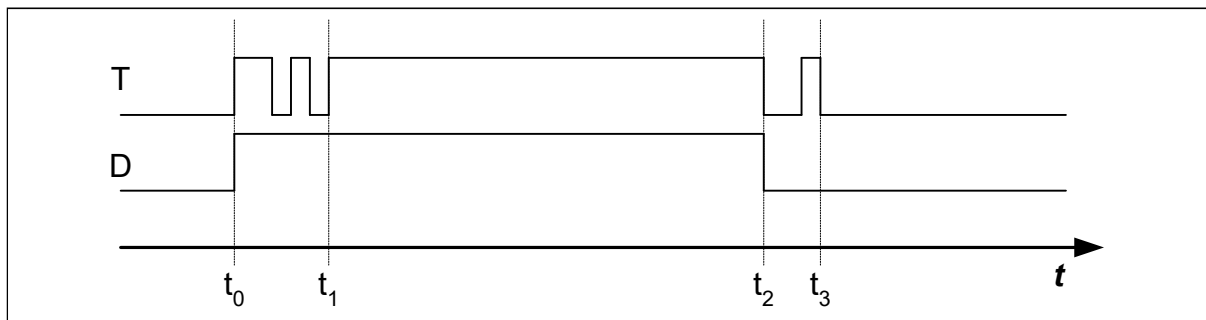


Abbildung 1: Prellendes Signal

Zum Zeitpunkt t_0 wird die Taste gedrückt und bis zum Zeitpunkt t_2 gedrückt gehalten. Der Taster schwingt jedoch noch unkontrolliert nach und erzeugt so bis t_2 bzw. t_3 noch Signalwechsel. Weder die Zahl der zusätzlichen Signalwechsel noch die zeitliche Länge des Nachschwingens sind konstant. In der Regel kann man nur für die maximale Zeitdauer t_{\max} bis zum Anliegen des Endwertes eine Angabe machen.

In dieser Situation bietet sich die Entprellung über einen Automaten an. Seine Taktfrequenz muss so hoch gewählt werden, dass auch das kürzestmögliche Ereignis für die jeweilige Anwendung sicher erfasst wird. Er hat dann die Aufgabe, das Signal T zu beobachten und erst dann einen Wechsel an einem Ausgang A zu signalisieren, wenn derselbe Wert für mindestens t_{\max} beobachtet wurde, d.h. der Taster sicher zur Ruhe gekommen ist.

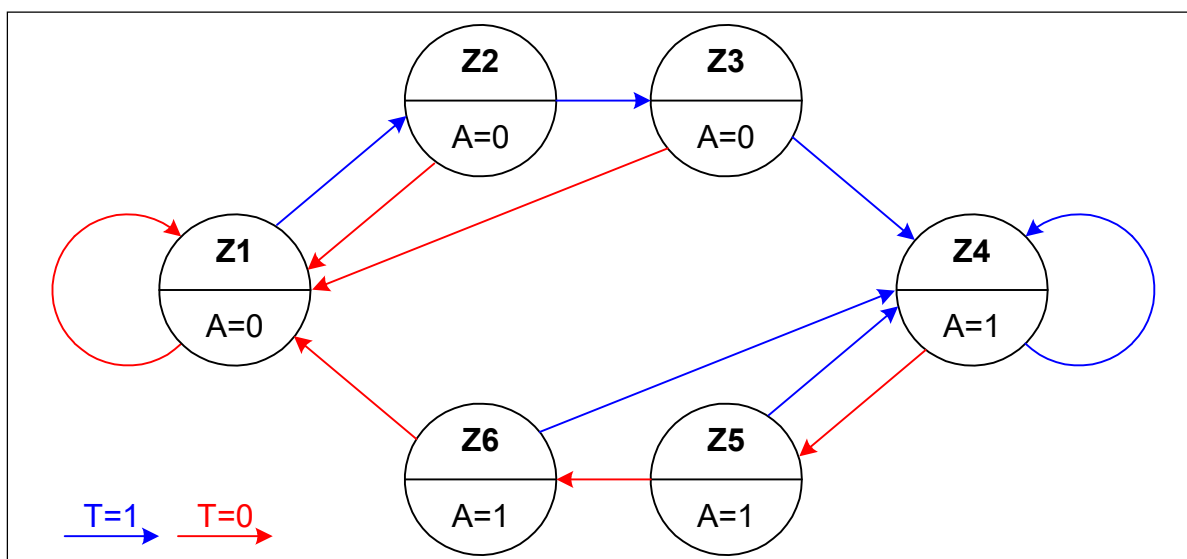


Abbildung 2: Entprellautomat

In Abbildung 2 ist das Zustandsübergangsdiagramm für einen solchen Automaten gezeichnet. Die Zeit t_{\max} sei hier sicher nach drei Taktperioden verstrichen. Wenn der Automat also dreimal *unmittelbar hintereinander* denselben Wert an seinem Eingang T gesehen hat, dann gibt er diesen Wert am Ausgang A aus.

In einem C-Programm in einer μC -Anwendung wäre T an einem Port angeschlossen und A wäre eine (globale) Variable, die vom Automaten (in einer ISR) verändert wird und von der Anwendung ausgewertet wird.

Die Zustände $Z1$ und $Z4$ sind die stationären Zustände (Taste sicher nicht gedrückt bzw. Taste sicher gedrückt), die Zustände $Z2$, $Z3$, $Z5$ und $Z6$ werden jeweils beim Signalwechsel durchlaufen.

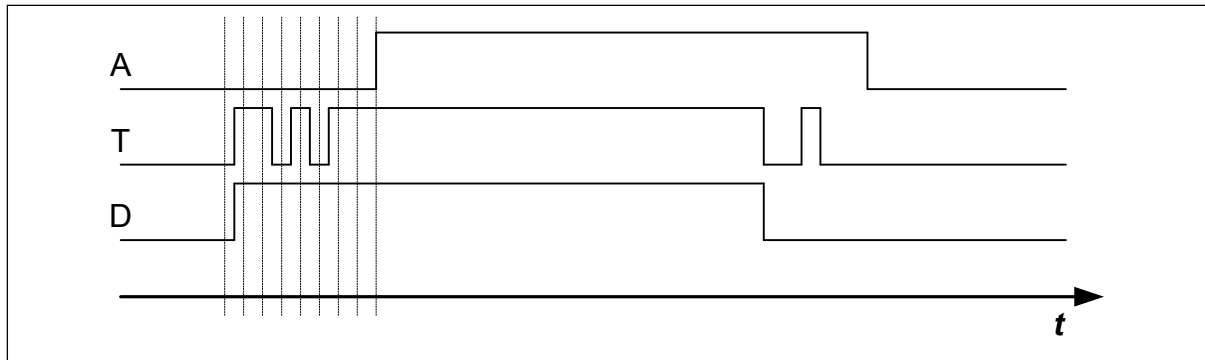


Abbildung 3: Entprelltes Signal

Passend zu dem Automaten aus Abbildung 2 zeigt Abbildung 3 den zeitlichen Verlauf des Ausgangssignals A . Die gestrichelt eingezeichneten Linien sind hier die Zeitpunkte, zu denen Automat jeweils aktiviert wird (Automatentakt). Sie setzen sich natürlich auch weiter in regelmäßigen Abständen fort.

Man erkennt hier gut die zeitliche Verzögerung des Ausgangssignals A gegenüber dem Eingangssignal T . Diese Verzögerung ist für Bedienelemente unerheblich (sie liegt typischerweise im Bereich von einigen Millisekunden). Für Signale, bei denen der genaue Zeitpunkt des ersten Wechsels entscheidend ist, ist diese Methode nicht geeignet.

Bemerkung

In einem μC -Programm würden die n Zustände ($Z2$, $Z3$ bzw. $Z5$, $Z6$), die durchlaufen werden müssen, bis der jeweilige Endzustand ($Z1$ bzw. $Z4$) erreicht wird, durch Zählerstände dargestellt. Damit spart man sich eine unnötige Aufzählung einzelner Zustände. Am Prinzip ändert sich damit nichts, es bleibt nach wie vor ein Automat.

```

T=Get_GPIO(...);           // Jetzigen Tastenwert einlesen
if ( A == T) z=MAX;
else
{
    if (z==0) A=T;         // Gespeicherten Tastenwert anpassen
    else z--;             // aber erst nach MAX Takten!
}

```

Listing 1: Automatenarstellung mit Zähler

Die Variable z enthält den derzeitigen Stand des Zählers. Der aktuell (sicher) erkannte Wert des Tasters (Signal A in Abbildung 3, Variable A in Listing 1: Automatenarstellung mit ZählerListing 1) bleibt solange erhalten, bis für MAX Takte nacheinander ein abweichender Wert T eingelesen wurde. Erst dann wird die Änderung als sicher eingestuft und A entsprechend aktualisiert.

Andere Programmteile nutzen A als Eingang (Wert der Taste), nicht T .

2 Beispiel 2: Drehgeber

In vielen Aufgabenstellungen ist man an Geschwindigkeit und Richtung einer Drehbewegung interessiert, wogegen die absolute Position (Winkel) des sich drehenden Elements unwichtig sind. Beispiele wären die Messung an einer Welle oder auch die Eingabe von Werten über Drehknöpfe. In solchen Fällen wird gerne ein Vier-Quadranten-Geber verwendet.

2.1 Aufbau des Gebers

Man teilt dazu eine Scheibe, die auf der sich drehenden Welle angebracht ist, in vier Quadranten ein (Abbildung 4, links).

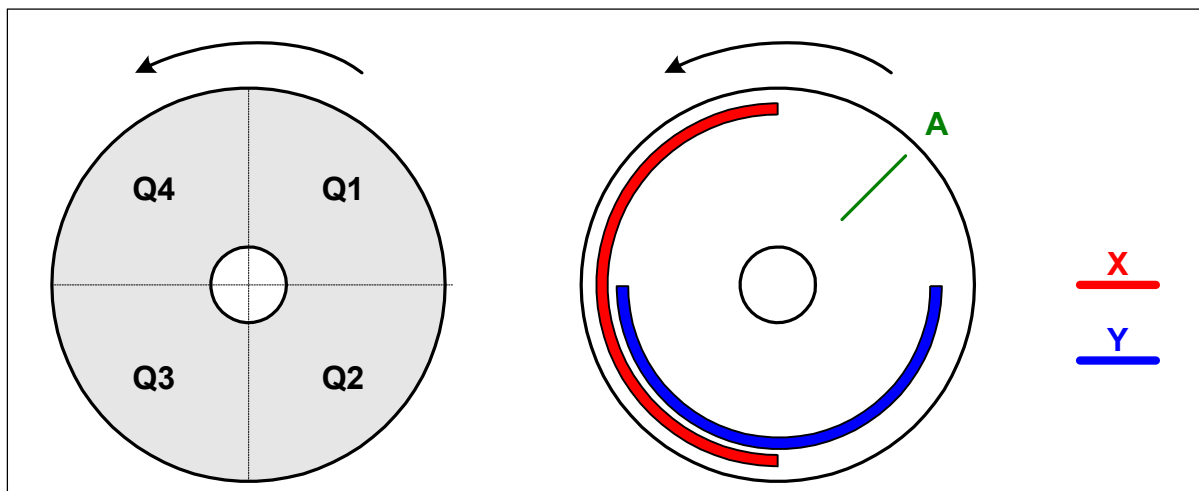


Abbildung 4: Prinzipieller Aufbau eines Vier-Quadranten-Gebers

Auf dieser Scheibe werden dann zwei konzentrische Spuren X und Y angebracht (Abbildung 4, rechts), die jeweils einen Halbkreis abdecken und gegeneinander um 90° versetzt sind. Diese Spuren werden mit einem Sensor A an einer festen Stelle abgetastet, d.h. der Sensor A ist ortsfest außerhalb der Scheibe befestigt. Ist die Spur vorhanden, wird beispielsweise eine 1 ausgegeben, sonst eine 0. Bei einfachen mechanischen Drehgebern können das Schleifringe sein, die über eine Bürste abgetastet werden, es können aber auch optische Spuren sein, die mit einer Lichtschranke abgetastet werden.

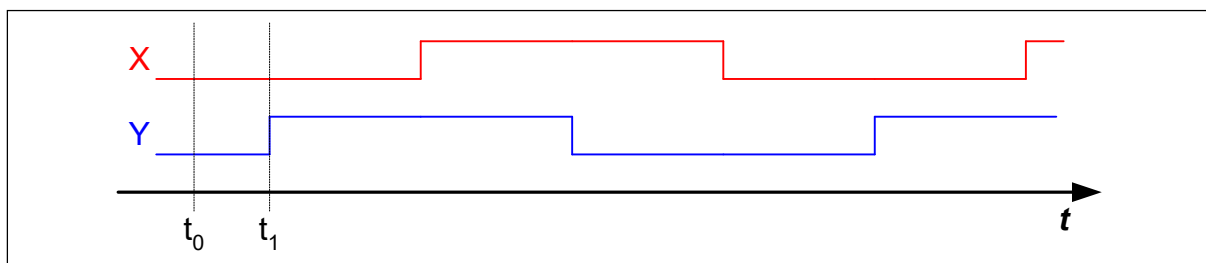


Abbildung 5: Signal des Gebers

Wenn sich beispielsweise die Scheibe aus Abbildung 4 in der gezeichneten Richtung dreht, dann wird der Sensor A das in Abbildung 5 gezeigte Signal liefern. Zum Zeitpunkt t_0 soll sich die Scheibe dabei in der rechts dargestellten Position befinden, zum Zeitpunkt t_1 hat sie sich eben um 45° gedreht. Nach einer einmaligen Umdrehung wiederholt sich der Signalverlauf. Aus diesem Grund kann man die Messung prinzipiell beliebig verfeinern, indem man die Scheibe in so viele Sektoren wie nötig einteilt und die Spuren X und Y innerhalb jedes Sektors

wie in Abbildung 4 rechts gezeigt anbringt. Die Sequenz Q1-Q4 wird dann eben in jedem Sektor aufs Neue durchlaufen.

2.2 Folgeautomat

Als Anwender möchte man aber in der Regel wissen, in welche Richtung (und eventuell wie schnell) sich das Element bewegt. Dazu kann man einen einfachen Automaten benutzen, der bei jedem erkannten Übergang von einem Quadranten zum nächsten ein Signal ausgibt und dazu die Richtung liefert.

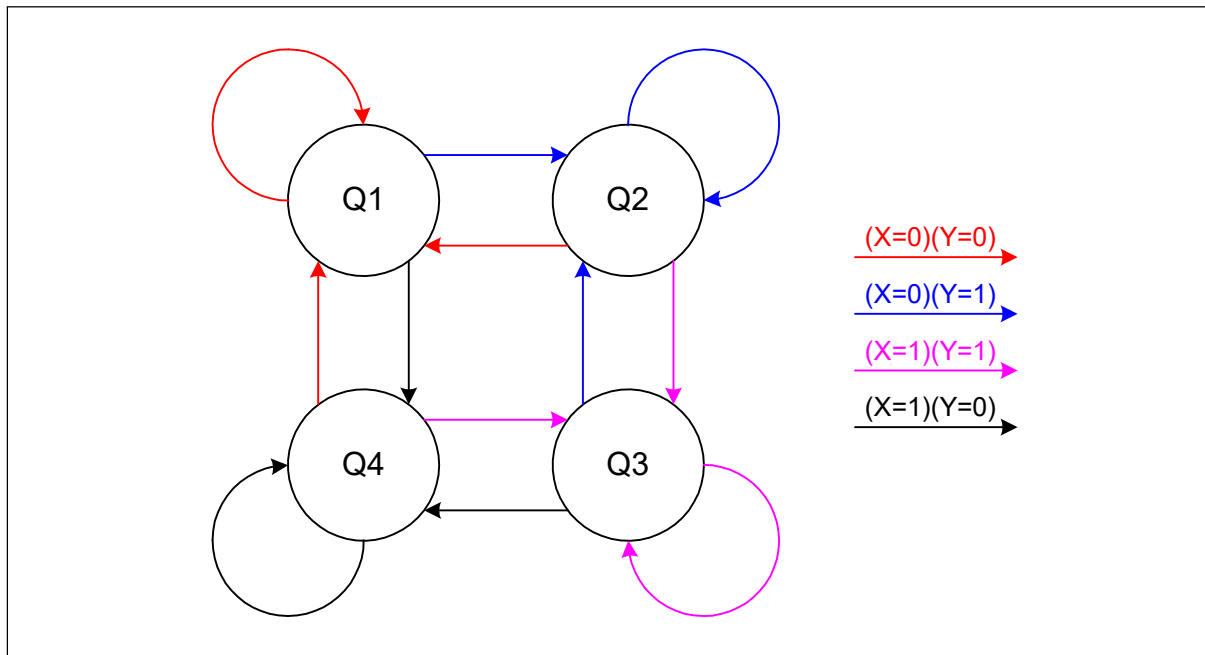


Abbildung 6: Folgeautomat

In Abbildung 6 ist das Zustandsdiagramm eines Automaten gezeichnet, der zunächst nur den Signalen folgt und sich damit jeweils im „richtigen“ Zustand für einen bestimmten Quadranten befindet. Wenn sich beispielsweise der Automat in Q1 befindet und die Signale X/Y melden, dass sich die Scheibe immer noch im Quadranten Q1 befindet, dann bleibt der Automat auch in Q1. Wenn (wie zum Zeitpunkt t_1 in Abbildung 5) der Geber mitteilt, dass sich jetzt der Quadrant Q2 unter dem Sensor befindet, dann wechselt auch der Automat nach Q2. Im Normalbetrieb sind nur die Übergänge zu erwarten, die in Abbildung 6 gezeigt sind. Allerdings ist keiner der Zustände vollständig, in Q1 fehlt z.B. der Übergang für $X \cdot Y$. Das wäre ein Sprung von Q1 nach Q3, der im Betrieb in zwei Fällen auftreten kann:

1. Einschalten des Automaten

Dabei könnte als erster Zustand Q1 gewählt worden sein, obwohl sich die Scheibe in Q3 befindet.

2. Scheibe läuft zu schnell (Automatentakt zu langsam)

Der Automat sieht die Eingänge ja nur zu den Taktzeitpunkten. Wenn ein Quadrant vollständig in einer Taktperiode unter dem Sensor durchläuft, dann verpasst der Automat den Quadranten.

In beiden Fällen kann keine Richtungsangabe mehr ausgegeben werden. Der Vollständigkeit halber müssen aber auch diese Übergänge vorgesehen werden (also hier vier weitere Übergänge Q1->Q3, Q2->Q4, Q3->Q1, Q4->Q2).

2.3 Auswerteautomat

Der bisherige Automat kann zwar den Quadranten bereits folgen, allerdings werden noch keine Ausgaben geliefert.

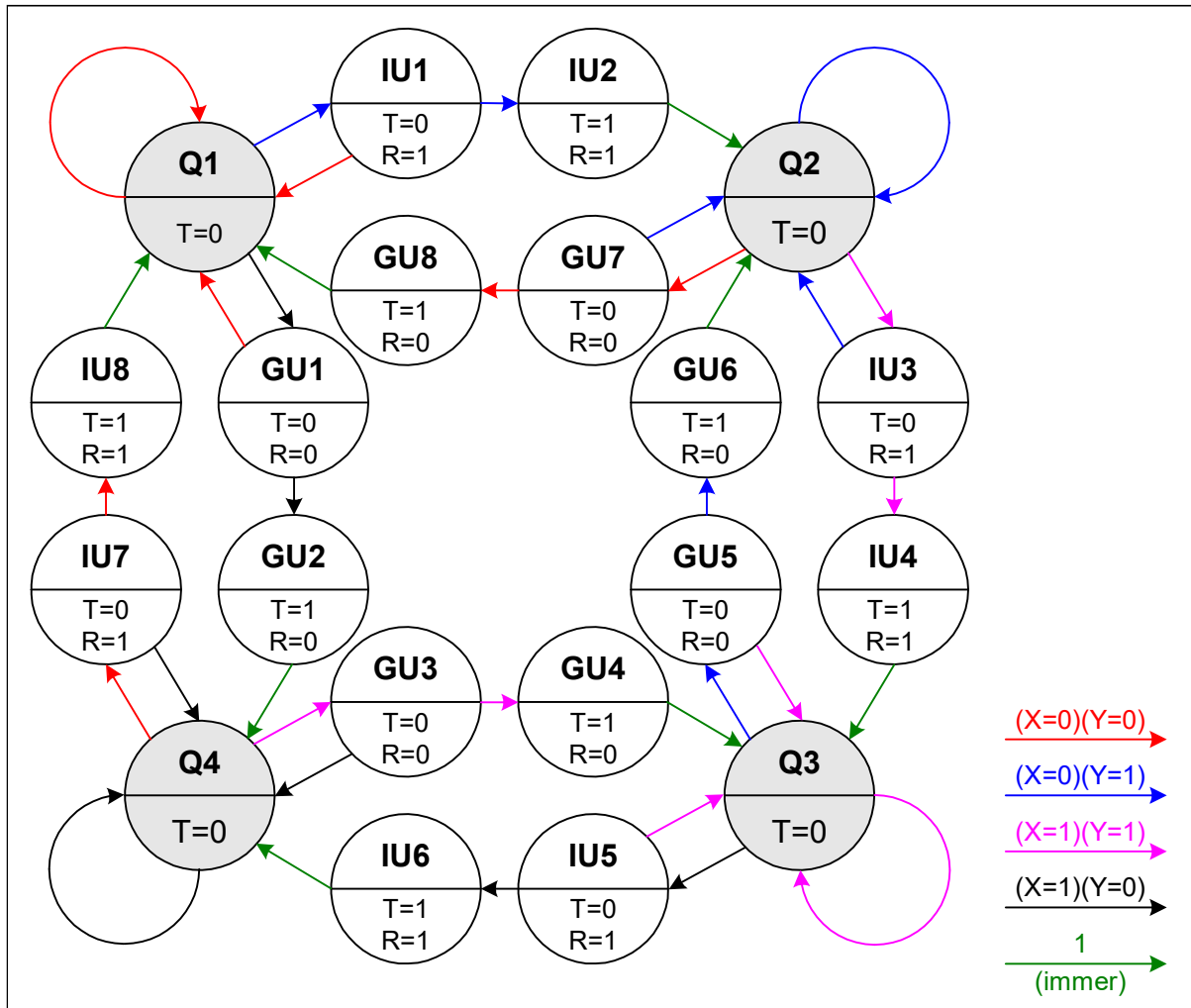


Abbildung 7: Auswerteautomat

In Abbildung 7 ist der Folgeautomat nur um Zwischenzustände erweitert worden, die beim Übergang von einem Quadranten zum nächsten durchlaufen werden. Auch wenn der Automat auf den ersten Blick komplex aussieht, so ist er doch sehr einfach zu verstehen. Die Signale T und R sind die Ausgabesignale des Automaten, wobei ein Übergang von 0 nach 1 bei T anzeigt, dass eine Drehbewegung festgestellt wurde (pro Übergang 0->1 ein Quadrantenwechsel). Die Drehrichtung wird mit dem Signal R angegeben, das also nur bei dem Übergang 0->1 von T gültig sein muss.

Die Geschwindigkeit der Drehbewegung ergibt sich also aus der Frequenz des Signals T (wozu man noch wissen muss, in wieviele Sektoren die Scheibe aufgeteilt wurde).

Der äußere Kreis der Zustände soll hier durchlaufen werden, wenn sich die Scheibe im Uhrzeigersinn dreht (daher die Namen IU). Der innere Kreis wird durchlaufen, wenn sich die Scheibe gegen den Uhrzeigersinn dreht. Für die Erklärung genügt also ein einziger Übergang von einem Quadranten zum nächsten in eine Richtung, weil alle anderen Übergänge genauso aufgebaut sind. Hier soll der Übergang von Q1 nach Q2 betrachtet werden.

Zunächst befindet sich noch Q1 unter dem Sensor, d.h. $\bar{X} \cdot \bar{Y}$. Entsprechend bleibt der Automat in Q1. Es kann keine Bewegung festgestellt werden (daher $T=0$). Über die Richtung kann nichts ausgesagt werden, sie ist ja ohne festgestellte Bewegung auch uninteressant (daher fehlt die Angabe zu R).

Meldet der Geber jetzt, dass sich Q2 unter dem Sensor befindet ($\bar{X} \cdot Y$), dann kann erstmals eine Richtung festgestellt werden. Entsprechend wird R vorbereitet ($R=1$). Allerdings wird *noch keine* Bewegung gemeldet, d.h. $T=0$. Der Grund ist, dass sich der Sensor zufällig genau über dem Wechsel von Q1 zu Q2 befinden könnte. Wenn die Scheibe dort stehenbleibt, dann ist die Auswertung von Y hier unsicher, d.h. es könnte zu manchen Zeitpunkten Y gemeldet werden und zu anderen Zeitpunkten \bar{Y} . Würde man dann jedesmal einen Wechsel anzeigen, dann würde die Scheibe zumindest in der Auswertung „zittern“, also eine Bewegung angezeigt werden, die gar nicht vorhanden ist. Im gegebenen Zustandsübergangsdiagramm führt dagegen eine solche unsichere Auswertung des Signals Y nur zu einem wiederholten Wechsel zwischen Q1 und IU1, ohne dass dabei jemals $T=1$ ausgegeben wird.

Erst wenn sich der Wechsel bestätigt wechselt der Automat in IU2. Hier wird zusätzlich zur Richtung auch noch die Bewegung angezeigt ($T=1$). Aus diesem Zustand wechselt jetzt der Automat *immer*, also ganz unabhängig von den Eingängen, in den Folgezustand Q2. Der Grund ist, dass zum einen das Signal T wieder zurückgesetzt werden muß (sonst könnte der nächste Übergang nicht wieder durch den Wechsel von 0->1 angezeigt werden) und zugleich für eine eventuelle Drehrichtungsänderung die Möglichkeit für den Übergang von Q2->Q1 über den inneren Kreis vorbereitet werden muß.

Sollte sich tatsächlich eine Änderung der Richtung genau dann ergeben, wenn sich der Automat in IU2 befindet, dann wird eben nur der Zustand Q2 zusätzlich durchlaufen, bevor der „Rückweg“ nach Q1 über GU7 und GU8 angetreten wird. Das schadet nicht weiter und man hat den Vorteil, dass ein eventuelles „Zittern“ wegen unsicherer Signalauswertung jetzt zu dem harmlosen Wechsel zwischen Q2 und GU7 führt, aber keine Bewegung über Wechsel an T vorgetäuscht wird.

Alle weiteren regulären Übergänge folgen demselben Muster. Die irregulären Übergänge (außerhalb der zu erwartenden Reihenfolge) können wie beim Folgeautomaten ergänzt werden, sie enden jeweils in einem der stabilen Zustände Q1-Q4.

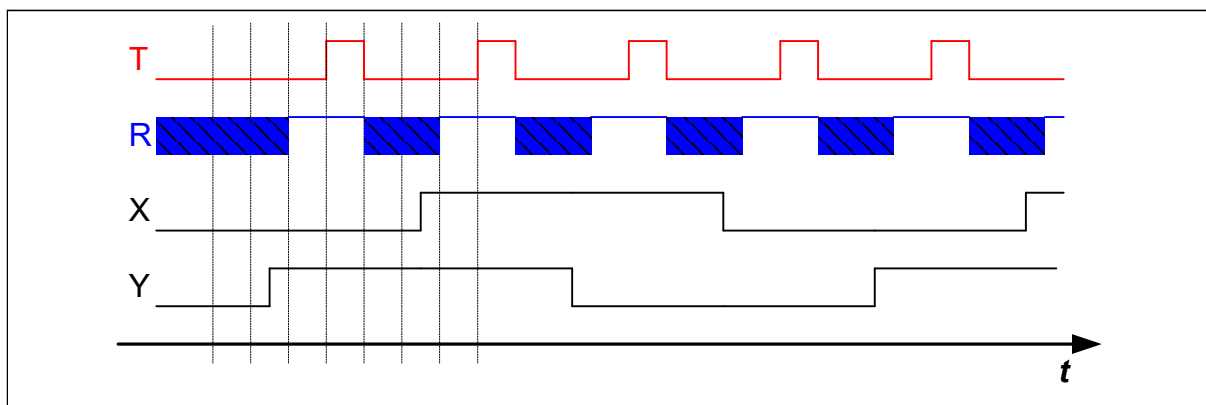


Abbildung 8: Ausgabesignale

In Abbildung 8 ist der zeitliche Verlauf der Ausgabesignale R und T für das Beispiel gezeigt. Der Taktzeitpunkt ist durch die gestrichelten Linien dargestellt. Die Schraffur bei R zeigt an, dass der Wert des Signals undefiniert ist. Anhand der Zeichnung ist zu erkennen, dass R vor dem Signalwechsel an T gültig wird und auch danach noch für einen Takt gültig bleibt. Zudem ist wieder der zeitliche Versatz der Ausgabe (bei T) bezüglich der Eingabe (X, Y) zu erkennen.