

1 Paritygenerate

In dieser Aufgabe soll ein Parity-Generate Modul entworfen werden, das für eine beliebige, serielle Eingangsbitfolge blockweise die Parität berechnet. Die Blockgröße soll 3 Bit betragen und die Ausgabe soll „0“ sein, wenn gerade Parität vorliegt, „1“ sonst. Die Ausgabe soll, nachdem ein 3-Bit Block vollständig ist, genau einen Takt lang anliegen, sonst soll „0“ anliegen

<i>Zeit</i>	0	1	2	3	4	5	6	7	8	9
<i>Bitfolge</i>	0	1	0	1	1	0	1	0	1	0
<i>Ausgabe</i>	0	0	0	1	0	0	0	0	0	0

- Erstellen Sie das Zustandsübergangsdiagramm einer Moore-Maschine, die die oben genannten Eigenschaften erfüllt.
- Implementieren Sie den Automaten und zeichnen Sie einen Schaltplan mit Gattern und FFs. Die Ausgabe soll durch eine Ausgangslogik realisiert werden.
- Implementieren Sie den Schaltplan auf dem FPGA. BTN0 steuert den Dateneingang, BTN1 das reset-Signal. Mit BTN3 wird der Automat getaktet.

2 Codeschloss

Entwerfen Sie eine Schaltung, die ein Codeschloss mit dem folgenden Verhalten darstellt:

- Durch drei Knöpfe (1, 2, 3) kann ein vierstelliger Code eingegeben werden. Nach der Eingabe der durch die Schaltung vorgegebenen richtigen Kombination (z. B. 1-3-2-1) soll sich das Schloss öffnen, was durch eine LED angezeigt wird.
- Es soll durch einen Reset-Knopf die Möglichkeit geben, eine nicht zu Ende eingegebene Kombination wieder zu löschen.
- Es soll verhindert werden, dass durch gleichzeitiges Drücken mehrerer Knöpfe der Code zufällig richtig eingegeben wird.

Das Verhalten der Schaltung kann durch eine FSM implementiert werden, bei der die Eingabe des Codes das Durchlaufen mehrerer Zustände bewirkt. Das Schloss gilt als geöffnet, wenn ein bestimmter Zustand erreicht wird.

- Zeichnen Sie das Zustandsdiagramm mit allen Übergängen.
- Setzen Sie Ihren Entwurf als Verilog-Modul um. Unter Umständen müssen Sie die Tasten der Platine entprellen, damit Sie das Codeschloss richtig benutzen können.