

1 Ampel

Entwickeln Sie einen *endlichen Automaten* (*finite state machine*, kurz FSM) der eine einfache Steuerung für eine Fußgängerampel realisiert. Die Ampel soll sich wie folgt verhalten:

- Wenn ein Fußgänger den Knopf drückt, fordert die Steuerung „rot“ für die Autos an und wartet bis die Ampel für die Autos auf „rot“ geschaltet hat.
- Nachdem die Autoampel rot geworden ist, schaltet die Fußgängerampel auf „grün“.
- Nach fünf Sekunden wird die Fußgängerampel wieder rot sowie die Autoampel grün und ist bereit für den nächsten Knopfdruck.

Dabei wird angenommen, dass es eine „Stoppuhr“ gibt, die von der Ampelsteuerung gestartet wird und ihr signalisiert, wenn fünf Sekunden vergangen sind.

Die Steuerung hat also (neben dem Takt- und Reset-Eingang) Eingänge für den Knopf, den die Fußgänger drücken (`knopf`), das Signal für die rote Autoampel (`fertig_auto_rot`) und das Signal der Stoppuhr (`fertig_5s`). Die Ausgänge der Steuerung sind die Anforderung für Rot (`start_auto_rot`), das Startsignal für die Stoppuhr (`start_5s`) und natürlich das Signal für die grüne Fußgängerampel (`gruen`).

- a) Zeichnen Sie den Automaten und realisieren Sie ihn in Verilog.
Wählen Sie die Kodierung der Zustände so, dass die Bits des Zustandsvektors direkt als Ausgang verwendet werden können.
In einem ersten Schritt steuern Sie die Eingänge `fertig_auto_rot` und `fertig_5s` per Button und zeigen Sie die Steuersignale `start_auto_rot` und `start_5s` auf LEDs an.
- b) Erstellen Sie nun einen Timer als separates Verilog-Module, der mit `start_5s` losläuft und nach fünf Sekunden das Signal `fertig_5s` erzeugt.
- c) Erstellen Sie ein weiteres Modul, das die Steuerung der Auto-Ampel (`start_auto_rot` und `fertig_auto_rot`) realisiert. Sie soll, wie in der Realität, dreifarbig schalten: grün -> gelb -> rot, dann rot -> rot+gelb -> grün mit je 1s Pause dazwischen.
Zeigen Sie die drei Farben auf drei LEDs an. Achten Sie darauf, dass zwischen dem Schalten der Auto- und der Fußgängerampel ebenfalls 1s verstreicht.