

Aufgabe 1

Eine Schaltung "statem" zur Steuerung eines Datenpfads hat die beiden Ausgänge "shift_o" und "load_o" und den Eingang "in_i".

```

1 architecture beh of tb is
2   signal clk, reset, ins, load, shift : std_ulogic;
3 begin
4
5 --
6 -- hier werden clk, reset und ins generiert
7 --
8
9 statem_i0 : statem
10 port map (
11   clk      => clk,
12   reset    => reset,
13   in_i     => ins,
14   load_o   => load,
15   shift_o  => shift
16 );
17 end architecture;
```

Die Schaltung "statem" soll verifiziert werden. In Abbildung 1 ist das Zustandsdiagramm abgebildet, nach dem sich die Schaltung "statem" verhalten soll. Die beiden Ausgänge sind als "load,shift" im Diagramm dargestellt, d.h. "1,0" bedeutet, dass load = "1" ist und shift = "0" ist.

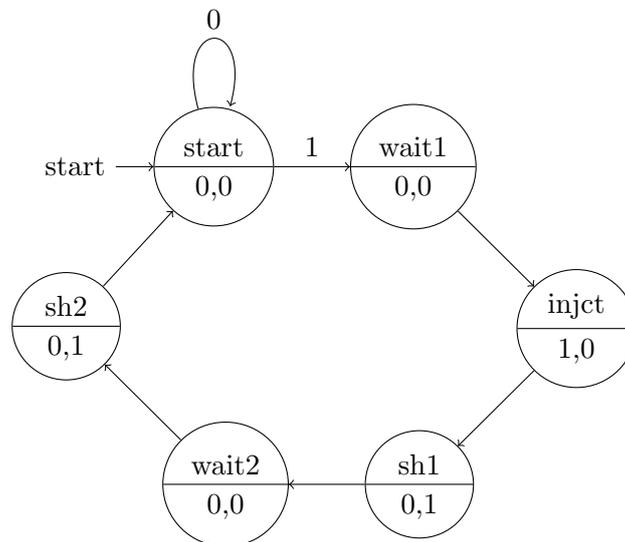


Abbildung 1: Zustandsdiagramm zur Spezifikation des Verhaltens der Schaltung "statem"

- Sie wissen nichts über die Implementierung des Automaten. Wie viele Zyklen benötigen Sie um nachzuweisen, dass die Implementierung sich genau gemäß des Zustandsdiagramms in Abbildung 1 verhält?
- Formulieren Sie eine assertion bezüglich der Signale "load" und "shift", die in jedem Takt gilt.

c) Wie viele Simulationszyklen benötigen Sie mindestens um jeden Zustand mindestens einmal erreicht zu haben und in jedem Zustand alle möglichen Eingangskombinationen ausprobiert zu haben? Zählen Sie die Zyklen ab der Deaktivierung des Resetsignals. Geben Sie eine Sequenz für das Eingangssignal "in_i" an, mit dem Sie diese minimale Anzahl von Zyklen erreichen können.

d) Nehmen Sie an, die Schaltung wurde mit folgender Eingangssequenz erfolgreich verifiziert: in_i = "11111110000001111111", d.h. die Schaltung hat sich für diese Sequenz genau gemäß des Zustandsdiagramms verhalten. Geben Sie ein Zustandsdiagramm für einen Automaten an, der sich für die Eingangssequenz genau gleich wie ein Automat gemäß des Zustandsdiagramms in Abbildung 1 verhält, der jedoch in folgenden Zyklen ein anderes Verhalten zeigen kann. Geben Sie eine Eingangssequenz für in_i an, bei der sich der Automat anders verhält.

e) Nehmen Sie an, dass Sie für die Verifikation auch den Zustand der Schaltung beobachten können. Begründen Sie, warum Ihnen dies bei der Verifikation helfen kann.

Aufgabe 2

Die Funktion `room_id` berechnet eine Identifikationsnummer für einen Raum aus den Elementen `room_number`, `floor` und `building`, die es für jeden Raum gibt. Die Daten der Räume sind in einem Array von structs gespeichert. Dieses Array hat 1000 Elemente, d.h. es gibt 1000 Räume. Nehmen Sie an, dass die Elemente des Arrays direkt nacheinander abgespeichert werden.

```
1 struct room_s {
2     int room_number;
3     int floor;
4     int building;
5 } rooms[1000];
6
7 int room_id(int room_index)
8 {
9     struct room_s *rp;
10
11     /* Compute the pointer to the room of index room_index */
12     rp = &rooms[0] + room_index;
13
14     /* Compute a room id number from room_number, floor and building number */
15     return (rp->room_number*1024 + rp->floor*16 + rp->building);
16 }
```

Nehmen Sie an Ihre Rechnerarchitektur hat einen Datencache mit einer Blockgröße von 16 Bytes und nehmen Sie an, dass das Array Cache Aligned ist, d.h. der Anfang des Arrays beginnt am Anfang eines Blocks. Nehmen Sie an, dass der Cache zu Beginn leer ist. Der Cache ist voll assoziativ und hat eine Größe von 1 kByte. Im Cache sind nur die Arraydaten.

a) Wie hoch ist die durchschnittliche Cache-Miss Rate in Bezug auf den Zugriff auf Elemente der Structs? Die Funktion `room_id` wird 10 mal mit unterschiedlichen Parametern aufgerufen. Die Parameterwerte sind gleichverteilt, d.h. die Wahrscheinlichkeit, dass `room_index` 87 ist, ist genauso groß wie die Wahrscheinlichkeit für irgendeinen anderen Wert. Nehmen Sie an, dass der Cache zu Beginn der 10 Zugriffe leer ist.

b) Nehmen Sie an die durchschnittliche Cache-Miss Rate des Datencache beträgt 50%. In Ihrem Algorithmus sind 20% aller Instruktionen load Instruktionen. Wenn die Daten im Cache vorhanden

Digitale Systeme 2 - Aufgaben Teil 3

sind, sind 2 Zyklen für den Zugriff notwendig. Wenn die Daten nicht im Cache sind, sind 40 Zyklen notwendig. Alle anderen Instruktionen benötigen 1 Zyklus. Berechnen Sie die durchschnittliche Anzahl der Zyklen pro Instruktion.

c) Nehmen Sie jetzt an, dass schon so viele Funktionsaufrufe von `room_id()` erfolgt sind, dass der Cache komplett gefüllt ist. Wie hoch ist die durchschnittliche Cache Miss Rate bei weiteren Aufrufen von `room_id()`?

d) Schlagen eine Änderung des structs `room_s` vor um die Cache Miss Rate zu minimieren. Wie groß ist die Cache Miss Rate mit Ihrem Vorschlag?