

Übungsblatt 4

(10 Punkte)

Präsenzaufgaben zur Woche ab Montag, 23. November 2015

Die praktischen Übungen finden im Raum OH16 / U09 statt. Die Bearbeitung der Aufgaben erfolgt in den Übungen.

Für die Bearbeitung dieses Blatts wird die virtuelle Maschine **ES-FPGA** verwendet.
 Bitte beachten Sie, dass darin abgelegte Daten bei einem Neustart *gelöscht* werden!

4.1 Ein Reaktionsspiel mit Zustandsautomat (5 Punkte)

In dieser Aufgabe implementieren Sie ein Reaktionsspiel für zwei Personen anhand des in Abbildung 1 gezeigten Zustandsautomaten. Zum Glück unterstützt Visual State VHDL nicht als Zielsprache, daher ist etwas Tümparbeit zur Umsetzung notwendig.

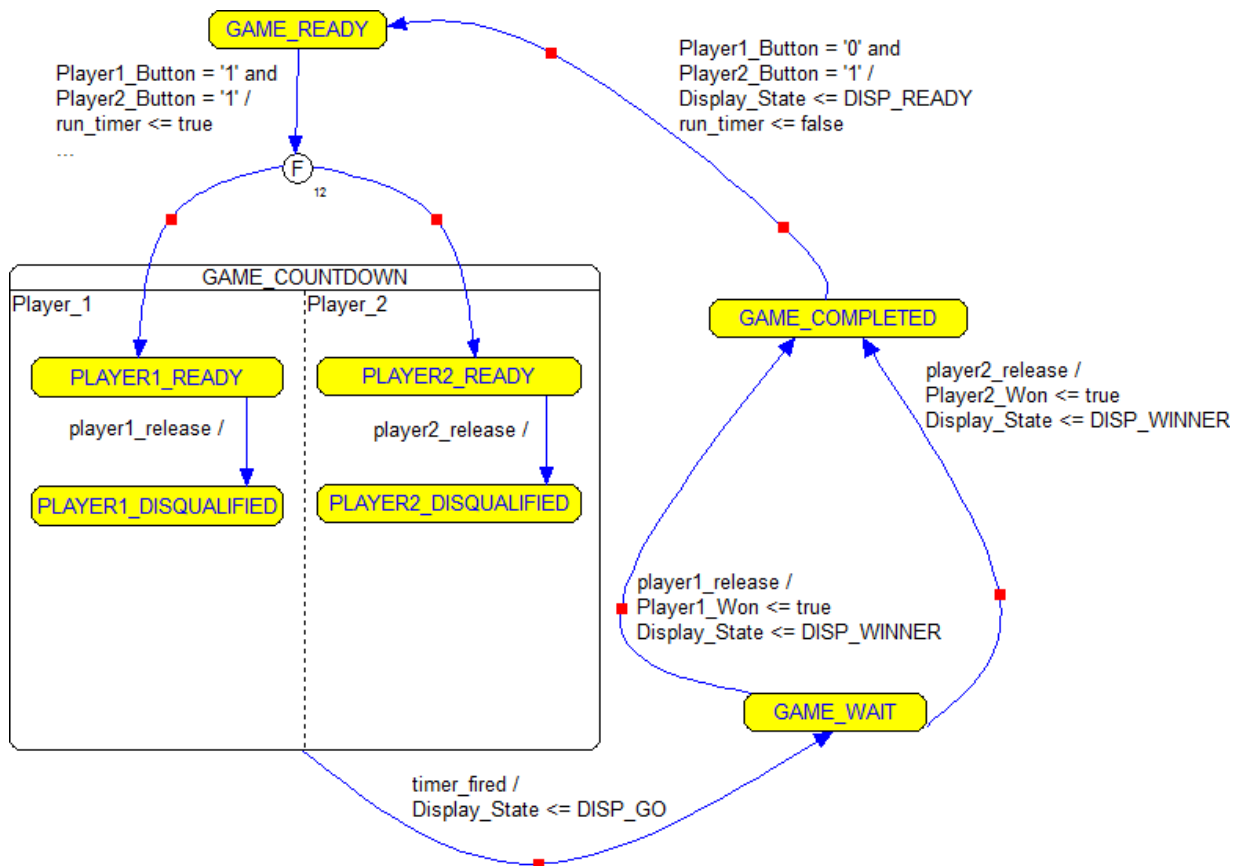


Abbildung 1: Zustandsdiagramm des Spiels

Regeln: Jedem Spieler ist eine Taste zugewiesen, die untere (*BTND*) für Spieler 1, die rechte (*BTNR*) für Spieler 2. Um ein Spiel zu starten, müssen beide Spieler ihre Taste betätigen. Nach einer zufälligen Wartezeit (3-10 Sekunden) wird "GO" auf dem Display angezeigt und die Spieler müssen versuchen, so schnell wie möglich ihre Taste loszulassen - der erste Spieler, der seine Taste loslässt gewinnt. Sollte ein Spieler jedoch die Taste loslassen, bevor "GO" angezeigt wurde, wird er disqualifiziert. Um ein neues Spiel starten zu können, müssen zunächst beide Spieler ihre Taste loslassen.

Aufgabe: Implementieren Sie den Zustandsautomaten für das Spiel in der Datei `GameLogic` aus der Vorgabe. Der Übergang von `GAME_READY` zu `GAME_COUNTDOWN` wurde bereits in der Vorlage als Beispiel mitgeliefert. Zur Implementierung der weiteren Zustandsübergänge müssen Sie in den Zweigen der `case`-Anweisung mittels `if ... then ... end if;` die notwendigen Trigger für jeden Zustandsübergang abfragen und die dadurch ausgelösten Aktionen und den Wechsel des aktuellen Zustands ausführen. Wenn im Diagramm nur ein Signalname wie z.B. `player1_release` genannt ist, handelt es sich um ein Signal vom Typ `Boolean`, welches direkt als Bedingung in eine `if`-Anweisung geschrieben werden kann.

Hinweis: Das Diagramm enthält (mindestens) einen Fehler, welchen Sie natürlich nicht in Ihre Implementierung übernehmen sollen.

Ein paar Notizen zur VHDL-Syntax:

- Boolesche Verknüpfungen verwenden Schlüsselworte statt Symbole, z.B. `not a`, `a and b`, `a or b`, `a xor b`, ...
- If-Anweisungen umschliessen immer komplette Blöcke, zusätzliche "Klammerung" wie z.B. in C oder Java ist nicht nötig:

```
1 if bool_ausdruck then
2   statements;
3 else
4   statements;
5 end if;
```

Der `else`-Teil kann natürlich auch entfallen, wenn er nicht benötigt wird.

- Ein Signal vom Typ `std_logic` ist kein in einer If-Anweisung verwendbarer Boolescher Ausdruck¹, es muss explizit mit einem Wert verglichen werden, z.B. mit `foo = '0'` oder `foo = '1'` (oder dem Ungleich-Operator `/=`). In dieser Aufgabe sollte das nur die Signale `Player1_Button` und `Player2_Button` betreffen.

Tip: Die LED-Leiste über den Schaltern zeigt für einfachere Fehlersuche die Ausgänge des `GameLogic`-Moduls an. Von links nach rechts zeigen die ersten vier LEDs den aktuellen Wert von `Display_State` (`DISP_READY`, `DISP_COUNTDOWN`, `DISP_GO`, `DISP_WINNER` und die letzten vier LEDs die Signale `Player1_Won`, `Player2_Won`, `Player1_Disq`, `Player2_Disq`.

4.2 Verbesserungen der Spielelogik (5 Punkte)

Wenn Sie das vorgegebene Zustandsdigramm nachimplementieren, gibt es zwei Probleme:

- Die Spieler können mogeln - wenn ein Spieler während der Wartezeit seine Taste loslässt und wieder drückt, wird er als disqualifiziert markiert, kann aber trotzdem noch gewinnen.
Sorgen Sie dafür, dass ein disqualifizierter Spieler nicht mehr als Gewinner akzeptiert wird.
- Wenn beide Spieler disqualifiziert wurden, wartet das Spiel trotzdem noch auf den Ablauf des Timers und eine neue Runde ist erst dann wieder startbar, wenn einer der Spieler seine Taste nochmals betätigt – oder evtl. gar nicht mehr startbar, wenn Sie den vorherigen Punkt bearbeitet haben. Da bei zwei disqualifizierten Spielern auch keiner mehr gewinnen kann, sollte an dieser Stelle lieber das Spiel abgebrochen werden.
Implementieren Sie einen Spielabbruch für den Fall, dass beide Spieler disqualifiziert wurden.

Allgemeine Hinweise: Die Übungstermine und weitere Informationen finden Sie über

<http://ls12-www.cs.tu-dortmund.de/daes/de/lehre/lehrveranstaltungen/wintersemester-20152016/es-1516.html>

¹Ein Signal vom Typ `boolean` dagegen schon