

Übungsblatt 8 – 22 Punkte

(Block 3 – insgesamt 92 Punkte)

Bearbeiten ab Samstag, 18. Dezember 2021.
Abgabe bis spätestens Donnerstag, 23. Dezember 2021, 23:59 Uhr.

8.1 Memory arrays in VHDL (22 Punkte)

Bisher hat es gereicht, kleine Speicher, die nur einige Bits speichern, mit Flip-Flops zu realisieren. Größere Datenspeicher werden durch Speicherarrays realisiert, die einen effizienteren Speicherzugriff erlauben.

In dieser Übung möchten wir euch näherbringen, wie solch ein Speicherarray realisiert werden kann. Betrachtet dazu den RAM-Baustein in Abbildung 1 mit Wortbreite 32 Bit. Die Funktionsteuerung des Bausteins erfolgt über den Eingang Read/Write (read). Über diesen kann man einstellen, ob geschrieben oder gelesen werden soll. Mit dem Adresseingang wird die Adresse des zu lesenden oder schreibenden Speicherelements festgelegt, wobei im Dateneingang die zu schreibenden Daten angelegt werden können. Der Out-Port gibt die Daten aus, die angefordert wurden.

Die restlichen Spezifikationen des Bausteins sind folgendermaßen definiert: Generisch einstellbare Speichergröße, Zugriffszeit 50 ns (Zeit, die man nach dem Anlegen einer Adresse warten muss, bis sie am Port out anliegen), Zykluszeit 100 ns (Zeit, die man nach dem Anlegen von zu schreibenden Daten warten muss, bevor ein neuer Zugriff möglich ist). Desweiteren ist in Abbildung 2 das Zeitverhalten für Lese- und Schreibzugriffe dargestellt.

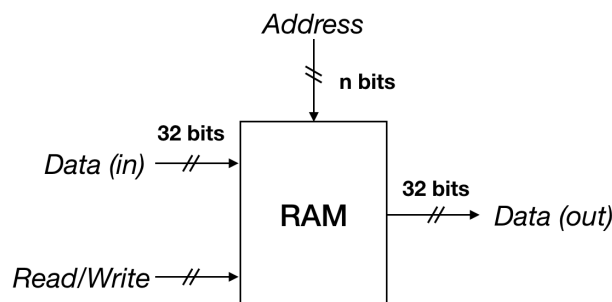


Abbildung 1: Verhaltens-Modell für einen RAM-Speicherbaustein.

Aufgaben:

Für den Entwurf des RAM-Bausteins in der beigefügten Datei "ram.vhdl" sollen folgende Schritte erfolgen:

- (2 Punkte) Wir benötigen zuerst eine Definition des Typen `BYTE_4`, der aus 4 Bytes (32 Bits) besteht. Erstellt hierzu ein Package, genannt `ram_pack`, welches diese Definition bereitstellt. Informationen zu Datentypen finden sich in Kapitel 4 im Skript.
- (2 Punkte) Der Nächste Schritt ist die Erstellung der Speichermatrix des RAMs. Wir benötigen dazu eine Definition basierend auf einem Feld-Typ (Array) mit Elementen `BYTE_4`. Erstellt diesen Feld-Typ, um mit ihm einen RAM-Speicher realisieren zu können.
- (4 Punkte) Die Lesefunktionalität des RAMs ist bereits in der Datei "ram.vhdl" implementiert. Studiert diese sorgfältig mit Hilfe der Abbildung 2 und erklärt, wie das Lesen in diesem Codesegment funktioniert. Bitte beachtet, dass Zugriffe auf den Speicher für das Lesen mit einem Adresswechsel beginnen.

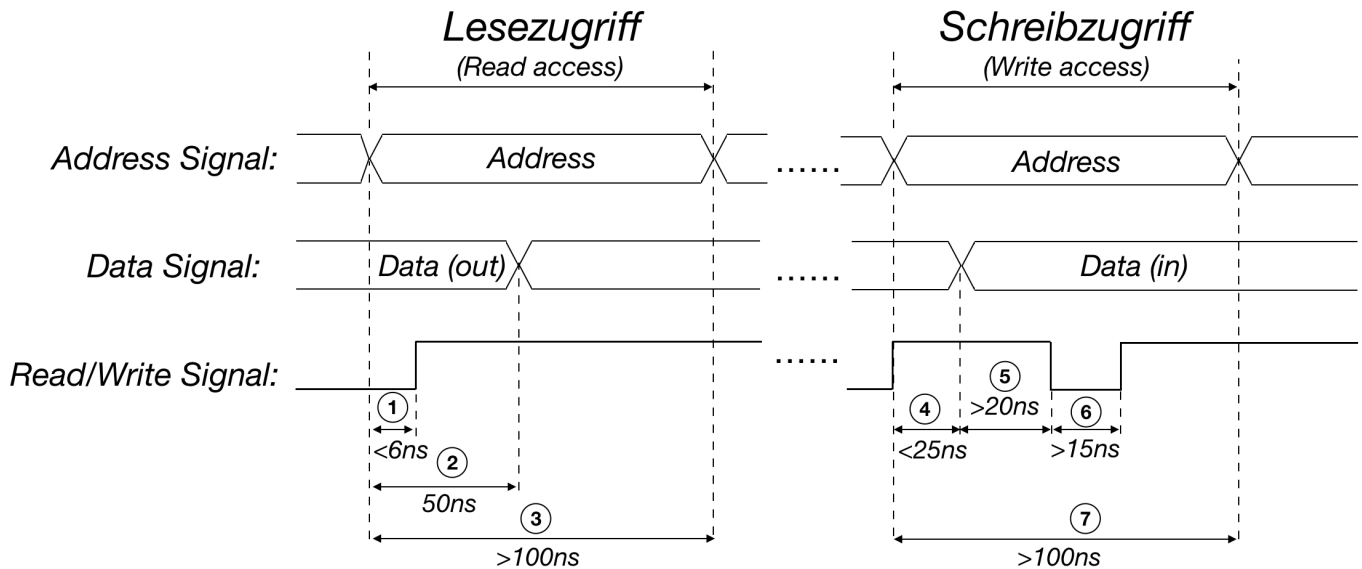


Abbildung 2: Zeitverhalten für Lesezugriff und Schreibzugriff.

- d. (4 Punkte) Implement nun die Schreibfunktionalität in einem weiteren Prozess. Beachtet, dass nur bei einer fallenden Flanke bei Read/Write die Schreibfunktion aktiviert wird. In dieser Teilaufgabe sollen die Timings vorerst nicht weiter beachtet werden. Testet euren RAM in einer Testbench durch das Speichern von zwei Unterschiedlichen Werten in zwei unterschiedlichen Adressen. Gebt die entsprechende Testbench bitte mit ab.
- e. (2 Punkte) In dieser Teilaufgabe geht es um die Einhaltung der Timings. Entwerft zwei zusätzliche Prozesse, welche die Einhaltung der Timings überprüfen. Dazu sollen Assert-Ausgaben angezeigt werden, wenn die Timing-Bedingungen verletzt werden. Hier sollen die Timings 100 ns für Zyklus (③) und ⑦ in Abbildung 2) und 15 ns für Schreibimpuls (⑥ in Abbildung 2) gelten.
- f. (4 Punkte) Es müssen im RAM-Baustein noch weitere Timing-Bedingungen beim Schreiben eingehalten werden. Implementiert diese ebenfalls in eurem RAM-Baustein, und gebt Assert-Meldungen aus, falls diese nicht eingehalten werden. Testet in der beigefügten Testbench, ob die Assert-Meldungen korrekt angezeigt werden.
- Bedingung 1: Während des Schreibprozesses liegen die Daten stabil an (d.h., gültige Werte).*
- Bedingung 2: Bevor Read/Write auf 0 sinkt, liegen Daten für mindestens 20 ns stabil an. (⑤ in Abbildung 2).*
- Bedingung 3: Nach dem Wechsel des Adresssignals sollten die Daten nicht länger als 25 ns gehalten werden, d.h. die neuen Daten müssen schnell genug nach einem Adresswechsel anliegen. (④ in Abbildung 2).*
- g. (2 Punkte) Erklärt, welche Arten von RAM es gibt und erläutert die jeweiligen Unterschiede sowie Vor- und Nachteile.
- h. (2 Punkte) Eine andere wichtige Art von Speicher ist ROM (read-only memory). Erläutert den Hauptunterschied zwischen ROM und RAM, und erklärt, welche Arten von ROM gibt es.