

Übungsblatt 8 – 22 Punkte

(Block 3 – insgesamt 90 Punkte)

Bearbeiten ab Samstag, 10. Juni 2023.
Abgabe bis spätestens Freitag, 16. Juni 2023, 23:59 Uhr.

8.1 Memory arrays in VHDL (18 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 Ihnen näherbringen, wie solch ein Speicherarray realisiert werden kann. Betrachten Sie dazu den RAM-Baustein in Abbildung 1 mit Wortbreite 32 Bit. Die Funktionsteuerung des Bausteins erfolgt über den Eingang Read/Write (read), der angibt ob gelesen oder geschrieben 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 zwischen zwei Adresswechseln liegen muss). Desweiteren ist in Abbildung 2 das Zeitverhalten für Lese- und Schreibzugriffe dargestellt.

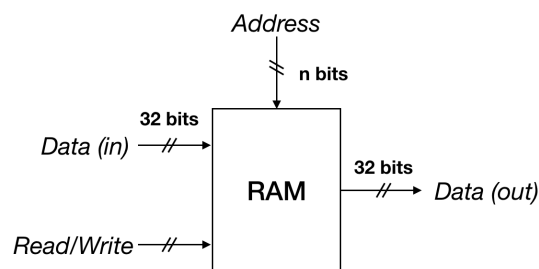


Abbildung 1: Verhaltens-Modell für einen RAM-Speicherbaustein.

Aufgaben:

Für den Entwurf des RAM-Bausteins in der beigelegten 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. Erstellen Sie 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*. Erstellen Sie diesen Feld-Typ, um mit ihm einen RAM-Speicher realisieren zu können.
- (2 Punkte) Die Lesefunktionalität des RAMs ist bereits in der Datei "*ram.vhdl*" implementiert. Studieren Sie diese sorgfältig mit Hilfe der Abbildung 2 und erklären Sie, wie das Lesen in diesem Codesegment funktioniert. Bitte beachten Sie, dass Zugriffe auf den Speicher für das Lesen mit einem Adresswechsel beginnen.
- (4 Punkte) Implementieren Sie die Schreibfunktionalität in einem weiteren Prozess. Beachten Sie, dass nur bei einer fallenden Flanke bei Read/Write die Schreibfunktion aktiviert wird. In dieser Teilaufgabe sollen die Timings vorerst nicht weiter beachtet werden. Testen Sie Ihren RAM durch das Speichern von zwei Unterschiedlichen Werten in zwei unterschiedlichen Adressen. Geben Sie die entsprechende Testbench bitte mit ab.

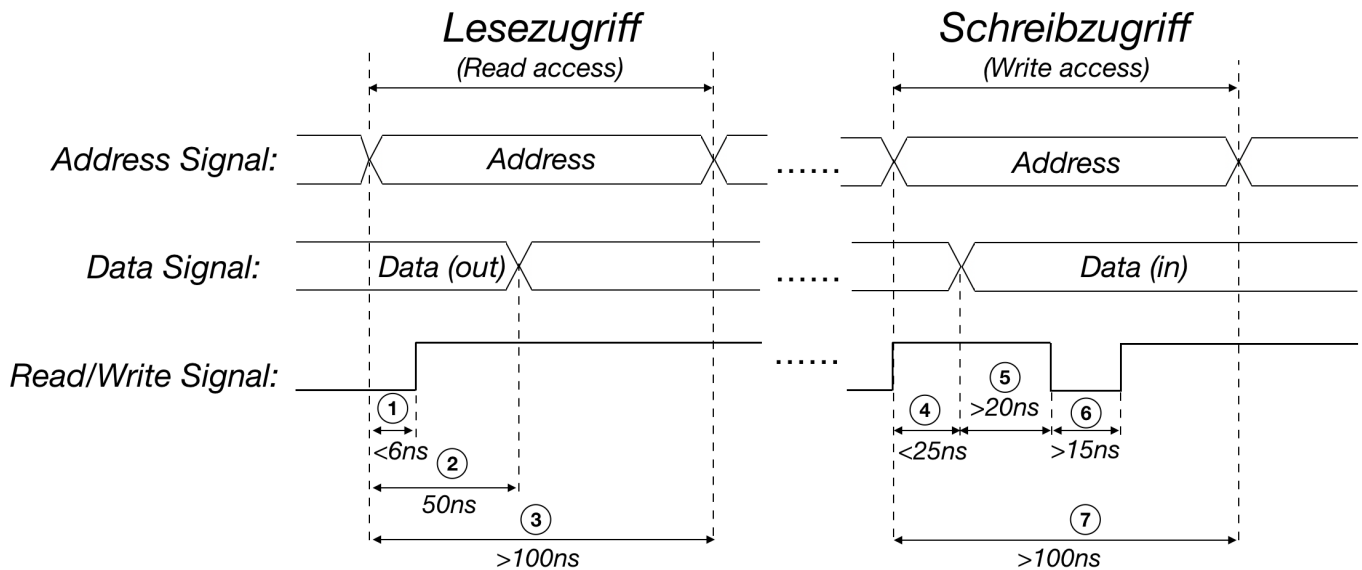


Abbildung 2: Zeitverhalten für Lesezugriff und Schreibzugriff.

- e. (4 Punkte) In dieser Teilaufgabe geht es um die Einhaltung der Timings. Entwerfen Sie 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. Implementieren Sie diese ebenfalls in der Schreibfunktionalität Ihres RAM-Bausteins, und geben Sie Assert-Meldungen aus, falls diese nicht eingehalten werden. Testen Sie in der beigelegten 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: Die zu schreibenden Daten dürfen nicht später als 25ns nach dem letzten Adresswechsel gewechselt werden, d.h. die neuen Daten müssen schnell genug nach einem Adresswechsel anliegen. (④ in Abbildung 2).

Beachten Sie dabei, dass die Bedingungen nur für das Schreiben gelten, d.h. Bedingung 2 und Bedingung 3 müssen nur überprüft werden wenn das Read/Write Signal auf 0 wechselt. Außerdem genügt es für Bedingung 1 zu überprüfen, ob die Daten zu Beginn und am Ende des Schreibprozesses identisch sind. Für Bedingung 3 genügt es außerdem, wenn die Daten vor dem Adresswechsel bereits anliegen. Dann wäre z.B. $-150 < 25$ wenn die Daten zum Zeitpunkt 100 geändert wurden und die Adresse zum Zeitpunkt 250. Die 20ns nach dem Adresswechsel müssen allerdings trotzdem eingehalten werden, d.h. wenn die Adresse zum Zeitpunkt 100 gewechselt wird und die Daten seit Zeitpunkt 95 anliegen würde ein Schreibprozess am Zeitpunkt 105 trotzdem Bedingung 2 verletzen.

8.2 Speicher (4 Punkte)

In dieser Aufgabe sollen kurz verschiedenen Arten von Speicher und deren Vor- und Nachteile betrachtet werden.

- a. (2 Punkte) Erklären Sie, welche Arten von RAM es gibt, geben Sie an wofür diese verwendet werden und erläutern Sie die jeweiligen Unterschiede sowie Vor- und Nachteile.
- b. (2 Punkte) Eine andere wichtige Art von Speicher ist ROM (read-only memory). Erläutern Sie den Hauptunterschied zwischen ROM und RAM, und erklären Sie, welche Arten von ROM gibt es.