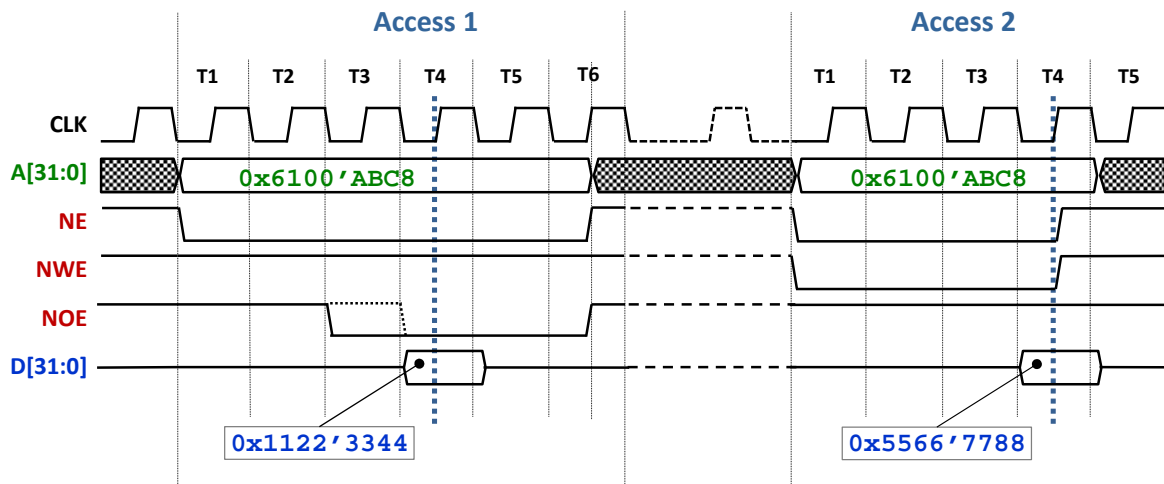


CT Übungsaufgaben

Microcontroller Basics

Aufgabe 1

Gegeben sind die folgenden Buszugriffe



- a) Geben Sie für beide Zugriffe jeweils die Richtung an (write oder read) sowie die Adresse des Zugriffs und den geschriebenen bzw. gelesenen Wert.

Access 1: read; 0x6100'ABC8 Adresse; 0x1122'3344 Daten gelesen

Access 2: Write; 0x6100'ABC8 Adresse; 0x5566'7788 Daten geschrieben

- b) Was enthält der Speicher vor dem Zugriff "Access 1"? Tragen Sie die Bytes, auf welche zugegriffen wird, mit ihren Adressen in der Memory Map ein. Der Prozessor ist little endian.

Adresse	Inhalt (Byte)
... C8	0x 44
... C9	0x 33
... CA	0x 22
... CB	0x 11

- c) Was enthält der Speicher nach dem Zugriff "Access 2"? Tragen Sie die Bytes, auf welche zugegriffen wird, mit ihren Adressen in der Memory Map ein. Der Prozessor ist little endian.

Adresse	Inhalt (Byte)
... C8	0x 33
... C9	0x 22
... CA	0x 66
... CB	0x 55

Aufgabe 2

Gegeben ist ein System Bus mit den 6 Adresslinien $A[5:0]$.

- a) Wie viele Bytes können damit adressiert werden?

$$2^6$$

- b) Unter wie vielen Adressen kann ein 8-bit Control Register angesprochen werden, wenn dafür 4 dieser Adressleitungen dekodiert werden?

$$2^{(6-2)} = 2^2 = 4 \text{ Bytes}$$

- c) Unter welchen Adressen (in Hex) kann das Control Register angesprochen werden, wenn nur die oberen 4 Adressleitungen wie folgt dekodiert werden:

select = $A[5] \& A[4] \& !A[3] \& !A[2]$

$$\begin{array}{cccccc} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{array} = \begin{array}{l} 0x30 \\ 0x31 \\ 0x32 \\ 0x33 \end{array}$$

$\Rightarrow 1100xx$

- d) Unter welchen Adressen (in Hex) kann das Control Register angesprochen werden, wenn nur die unteren 4 Adressleitungen wie folgt dekodiert werden:

select = $!A[3] \& A[2] \& A[1] \& !A[0]$

$$\begin{array}{cccc} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{array} = 0x06, 0x16, 0x26, 0x36$$

- e) Unter welchen Adressen (in Hex) kann das Control Register angesprochen werden, wenn nur die mittleren 4 Adressleitungen wie folgt dekodiert werden:

select = $!A[4] \& !A[3] \& A[2] \& !A[1]$

$$\begin{array}{cccc} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array} = \begin{array}{l} 0x04 \\ 0x05 \\ 0x14 \\ 0x15 \end{array}$$

- f) Wie müssen die Adressen dekodiert werden, wenn das Control Register genau unter der Adresse $0x28$ angesprochen werden soll?

$$0x18 = (00)10 \cdot 1000$$

$$A[5] \wedge !A[4] \wedge A[3] \wedge !A[2] \wedge !A[1] \wedge A[0]$$

Aufgabe 3

Schreiben Sie Codesequenzen in C für die folgenden Fälle. Verwenden Sie unsigned integer Typen aus `stdint.h`

- a) Lesen Sie den Wert eines 8-bit Control Registers an der Adresse `0x6100'0007` in eine von Ihnen zu definierende Variable ein.

```
#define REG (*(volatile uint8_t *) (0x6100'0007))  
uint8_t my-var  
my-var = REG
```

- b) Setzen Sie sämtliche Bits eines 16-Bit Control Registers an Adresse `0x6100'0008` auf ,1'.

```
#define REG (*(volatile uint16_t *) (0x6100'0008))  
REG = 0xFFFF
```

- c) Warten Sie in einer Schleife, bis Bit 15 im 32-bit Control Register an der Adresse `0x6100'000C` auf ,1' gesetzt ist.

```
#define REG (*(volatile uint32_t *) (0x6100'000C))  
while (! (REG & 0x8000'0000))
```

- d) Setzen Sie Bit 16 im Control Register an Adresse `0x6100'0010` auf ,1' ohne die anderen Bits des Registers zu verändern.

```
#define REG (*(volatile uint32_t *) (0x6100'0010))  
REG |= 1'0000'0000'0000'0000
```