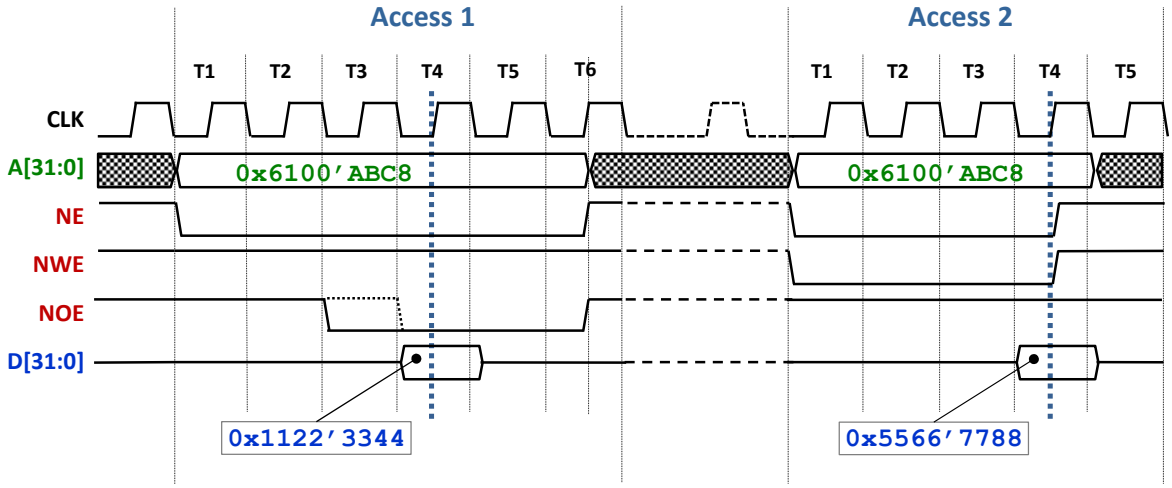
CT Übungsaufgaben
Microcontroller Basics

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 von Adresse 0x6100'ABC8 mit Wert 0x1122'3344
access 2: write nach Adresse 0x6100'ABC8 mit Wert 0x5566'7788

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)
0x6100'ABC8	0x44
0x6100'ABC9	0x33
0x6100'ABCA	0x22
0x6100'ABCB	0x11

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)
0x6100'ABC8	0x88
0x6100'ABC9	0x77
0x6100'ABCA	0x66
0x6100'ABCB	0x55

Aufgabe 2

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

a) Wie viele Bytes können damit adressiert werden?

2^6 = 64 → Es können 64 Bytes adressiert werden.

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

$2^{(6-4)}$ = 4 → Es können 4 Bytes adressiert werden.

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]
```

1100XXb → 0x30, 0x31, 0x32, 0x33

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]
```

XX0110b → 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]
```

X0010Xb → 0x04, 0x05, 0x24, 0x25

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

```
select = A[5] & !A[4] & A[3] & !A[2] & !A[1] & !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 MY_BYTE_REG   (*((volatile uint8_t *)(0x61000007)))

uint8_t my_var;

my_var = MY_BYTE_REG;
```

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

```
#define MY_HALFWORD_REG (*((volatile uint16_t *)(0x61000008)))

MY_HALFWORD_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 MY_WORD_REG   (*((volatile uint32_t *)(0x6100000C)))

while (!(MY_WORD_REG & 0x00008000)){
}
```

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

```
#define MY_WORD_REG2   (*((volatile uint32_t *)(0x61000010)))

MY_WORD_REG2 |= 0x00010000;
```