

Mikroprozessorpraktikum WS 2011/12  
Aufgabenkomplex: 2

Teilnehmer:

Marco Träger, Matr. 4130515  
Alexander Steen, Matr. 4357549

Gruppe: Freitag, Arbeitsplatz: HWP 1

## A 2.1 Taktfrequenz

### A 2.1.1 Bestimmen Sie messtechnisch die Frequenz der LFXT1CLK- und XT2CLK-Taktquelle.

Zur Lösung dieser Aufgabe werden zusätzlich zu den in Aufgabenblock 1 beschriebenen Registern das BC2CTL2-Register benutzt. In diesem Register werden einige der Bits für die Taktsteuerung verwaltet. Die zwei höchstwertigsten Bits legen die Taktquelle und die zwei nächst niederen Bits die Auswahl des Taktteilers fest. Für diese Aufgabe wird die Taktteilung ausgeschaltet (also auf den Divisor 1), damit der tatsächliche Takt gemessen wird.

**LFXT1CLK** Um auf P5.4 das Taktsignal messen zu können, wird nach dem Verbinden der Portleitung mit dem Takt (via P5SEL) das Messgerät angeschaltet. Nun wird das folgende Programm ausgeführt:

```
void aufgabe211() {
    P5SEL |= (1 << 4); // Leitung mit Modul verbinden
    P5DIR |= (1 << 4); // Als Ausgang fungieren
    BC2CTL2 = (BC2CTL2 & ~SELM_3) | SELM_3; // LFXTCLK
    BC2CTL2 = (BC2CTL2 & ~DIVM_3) | DIVM_0; // Divisor 1
}
```

Am Messgerät lässt sich eine Frequenz von 32,76927 kHz messen. Dies ist die Taktfrequenz der LFXT1CLK.

**XT2CLK** Diese Messung wird analog zur LFXT1CLK-Messung durchgeführt. Einzig die Selektion der Taktquelle ändert sich (siehe Code):

```
void aufgabe211() {
    P5SEL |= (1 << 4); // Leitung mit Modul verbinden
    P5DIR |= (1 << 4); // Als Ausgang fungieren
    BC2CTL2 = (BC2CTL2 & ~SELM_3) | SELM_2; // XT2CLK
    BC2CTL2 = (BC2CTL2 & ~DIVM_3) | DIVM_0; // Divisor 1
}
```

Aus der Messung ergibt sich eine Taktfrequenz von 7,373165 MHz für die XT2CLK-Taktquelle.

### A 2.1.2 Bestimmen Sie messtechnisch die minimale und maximale Taktfrequenz des MCLK-Taktes, die sich auf Basis der LFXT1CLK-, XT2CLK- und DCOCLK-Taktquellen bereitstellen lässt. Belegen Sie die Messergebnisse mit einer Berechnung auf Basis aller Komponenten aus den Blockschaltbildern.

**LFXT1CLK** Als maximale Frequenz dieser Taktquelle kann das Ergebnis aus Aufgabe 2.1.1 hergenommen werden, also  $f_{max} \approx 32,76927 \text{ kHz}$ . Durch Modifikation mit dem größten Taktteiler (ein Achtel) können wir hier den minimalen Takt  $f_{min}$  produzieren. Also ergibt sich als Code:

```
void aufgabe212() {
    P5SEL |= (1 << 4); // Leitung mit Modul verbinden
    P5DIR |= (1 << 4); // Als Ausgang fungieren
    BC2CTL2 = (BC2CTL2 & ~SELM_3) | SELM_3; // LFXTCLK
    BC2CTL2 = (BC2CTL2 & ~DIVM_3) | DIVM_3; // Divisor 8
}
```

Aus der Messung ergibt sich ein Takt von 4,096158 kHz.

Dies deckt sich mit den Erwartungen, da sich für die minimale Taktfrequenz rechnerisch ergibt:

$$f_{min} = f_{max} \cdot \frac{1}{8} \approx 32,76927 \text{ kHz} \cdot \frac{1}{8} \approx 4,096159 \text{ kHz}$$

**XT2CLK** Analog zur LFX1CLK-Taktquelle, können wir die maximale Frequenz aus Aufgabe 2.1.1 nehmen, die minimale Frequenz wiederum durch den Takteiler mit Divisor 8 erreichen. Der Code dazu ist ebenfalls analog, wird darum nicht wiederholt.

Also ergibt sich für die maximale Frequenz  $f_{max} \approx 7,373165$  MHz.

Die Messung des minimalen Taktes ergibt eine Frequenz von 921,6456 kHz.

Auch dies stimmt mit dem rechnerischen Ergebnis überein:

$$f_{min} = f_{max} \cdot \frac{1}{8} \approx 7,373165 \text{ MHz} \cdot \frac{1}{8} \approx 921,6456 \text{ kHz}$$

**DCOCLK** Um die minimale und maximale Frequenz der DCOCLK-Taktquelle zu messen, binden wir in der `main.c` wieder die DCO-Quelle ein (Funktionsaufruf `DCO()`). Bei dieser Taktquelle können wir zusätzlich mit Hilfe des `DCOR`-Bits die Frequenz manipulieren. Dies kann ebenfalls im `BCSCTL2`-Register gesetzt werden. Die erste Messung wurde mit `DCOR = 0` durchgeführt:

```
void aufgabe212() {
    P5SEL |= (1 << 4); // Leitung mit Modul verbinden
    P5DIR |= (1 << 4); // Als Ausgang fungieren
    BCSCTL2 = (BCSCTL2 & ~SELM_3) | SELM_0; // DCO
    BCSCTL2 = (BCSCTL2 & ~DIVM_3) | DIVM_0; // Divisor 1
    BCSCTL2 &= ~DCOR; // DCOR auf 0
}
```

Die Messung ergibt einen Takt von  $f_{DCOR=0} \approx 1,7026$  MHz. Mit gesetztem `DCOR`-Bit ergibt die Messung  $f_{DCOR=1} \approx 7,3684$  MHz, was also die maximale Frequenz  $f_{max}$  ist.

Für die Messung der minimalen Frequenz wird der Takteiler wie in den vorigen Aufgaben genutzt, deshalb wird der Code nicht gezeigt. Die Messung für die minimale Frequenz ergibt einen Takt von ca. 212,8 kHz. Dies bestätigt das rechnerische Ergebnis:

$$f_{min} = f_{DCOR=0} \cdot \frac{1}{8} \approx 1,7026 \text{ MHz} \cdot \frac{1}{8} = 212,825 \text{ kHz}$$

**A 2.1.3** An P2.5 ist ein Oszillatorwiderstand  $R_{osc}$  von 39kOhm angeschlossen. Erläutern Sie, wie der externe Widerstand für den DCOCLK-Taktgenerator nutzbar gemacht wird.

Der externe Oszillatorwiderstand kann mit Hilfe des `DCOR`-Bits über das `BCSCTL2`-Register dazugeschaltet werden. Ist das Bit auf Eins geschaltet, wird der externe Widerstand genutzt.

**A 2.1.4** Welchen Einfluss hat der Widerstand auf den DCOCLK-Taktgenerator?

Bei der Auswahl des Oszillatorwiderstands via `DCOR`-Bit wird die Taktfrequenz größer. Wie man aus der Tabelle aus Aufgabe A 2.2.1 entnehmen kann, ist bei gleichem Divisor der Takt mit gewähltem Oszillatorwiderstand (ganz grob) vier Mal größer als ohne.

Dies wird wohl daher kommen, dass der Oszillatorwiderstand zusätzlich zum Taktgenerator schwingt und mit dem Takt positiv interferiert, sodass sich beide Takte zu einem größeren Takt ergänzen.

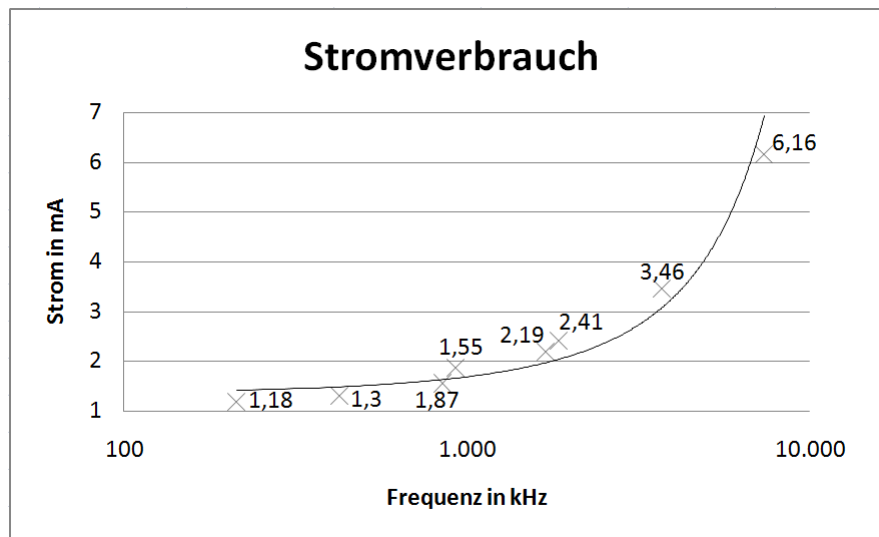
## A 2.2 Stromverbrauch

**A 2.2.1** Der MCLK-Takt soll durch den DC0CLK-Taktgenerator bereitgestellt werden. Ermitteln Sie für diesen Fall die Abhängigkeit des Stromverbrauchs von der Taktfrequenz. Stellen Sie die Abhängigkeit für einen Bereich von 100kHz bis 10MHz grafisch dar.

Die gemessenen Ströme sind wie folgt:

DC0R	Divisor	Frequenz [Hz]	Strom [mA]
0	8	$212,49 \cdot 10^3$	1,18
0	4	$426,35 \cdot 10^3$	1,30
0	2	$852,32 \cdot 10^3$	1,55
0	1	$1,703 \cdot 10^6$	2,19
1	8	$926,74 \cdot 10^3$	1,87
1	4	$1,852 \cdot 10^6$	2,41
1	2	$3,702 \cdot 10^6$	3,46
1	1	$7,371 \cdot 10^6$	6,16

Dann ergibt sich als Abhängigkeit zwischen Frequenz und Stromverbrauch:



Die durchgezogene Linie stellt eine exponentielle Ausgleichskurve dar.

## A 2.3 Taktumschaltung

**A 2.3.1** Entwickeln Sie ein Programm, das auf Tastendruck die Taktfrequenz des Mikrocontrollers zwischen 4,096kHz und 7,3728 MHz umschaltet.

Das nachfolgende Programm schaltet bei einem Druck auf die rechte Taste die Taktfrequenz auf 4,096 kHz und bei einem Druck auf die linke Taste auf 7,3728 MHz.

```

1 // Initialisierung der Register
2 // wird nicht in der while-Schleife
3 // aufgerufen.
4 init231() {
5     // alle LEDs ausschalten
6     // und Register korrekt setzen
7     P4SEL &= ~(0x07); // I/O Funktion
8     P4DIR |= 0x07; // Output
9     P4OUT |= 0x07; // Alle LEDs
10    LEDOFF; // aus
11
12    // Schalter einrichten
13    P1SEL &= ~(0x03); // I/O Funktion
14    P1DIR &= ~(0x03); // Input
15
16    // Taktfrequenzen erreichbar:
17    // 4,096kHz mit LFXTCLK Divisor /8
18    // 7,3728 MHz mit DCO Divisor /1 und DCOR 1
19
20    // Output des Takts
21    P5SEL |= (1 << 4);
22    P5DIR |= (1 << 4);
23 }
24 // Hauptprogramm
25 void aufgabe231() {
26     if (P1IN & 0x01) // rechte Taste gedrueckt
27     {
28         BCSCTL2 = (BCSCTL2 & ~SELM_3) | SELM_3; // LFXTCLK
29         BCSCTL2 = (BCSCTL2 & ~DIVM_3) | DIVM_3; // Divisor 8
30     }
31     if (P1IN & 0x02) // linke Taste gedrueckt
32     {
33         // DCO
34         BCSCTL2 = (BCSCTL2 & ~SELM_3) | SELM_0; // DCO
35         BCSCTL2 = (BCSCTL2 & ~DIVM_3) | DIVM_0; // Divisor 1
36         BCSCTL2 |= DCOR; // DCOR auf 1
37     }
38 }

```

Bei den Messungen ergibt sich für die Frequenz 7,347 MHz ein Stromverbrauch von ca. 4,87 mA. Bei einer Taktfrequenz von 4,096 kHz wurde ein Stromverbrauch von ca. 0,491 mA gemessen.

**A 2.3.2** Welche Schlußfolgerungen hinsichtlich des Energieverbrauches ziehen Sie? Berechnen Sie für beide gemessenen Stromverbrauchswerte die theoretisch mögliche Batterielaufzeit des MSB430H bei Nutzung einer Batterie mit einer Kapazität von 1100mAh.

Da der Energieverbrauch bei höheren Taktfrequenzen schnell ansteigt, sollte man den Takt nach Möglichkeit gering halten, um so die Batterielaufzeit zu optimieren. Die Lebensdauer  $t$  einer Batterie lässt sich durch die Formel

$$t = \frac{\text{Kapazität}}{\text{Verbrauch}}$$

bestimmen.

Bei einer Batterie mit einer Kapazität von 1100 *mAh* ergibt sich dann für einen Takt von 4,096 kHz:

$$t = \frac{1100 \text{ mAh}}{0,491 \text{ mA}} = 2240 \text{ h} \approx 3,07 \text{ Monate}$$

Und für den Takt von 7,347 MHz:

$$t = \frac{1100 \text{ mAh}}{4,87 \text{ mA}} = 225,9 \text{ h} \approx 9,41 \text{ Tage}$$

Bei der niedrigeren Frequenz kann das Modul also ca. drei Monate mit einer Batterieladung auskommen, bei der höheren Frequenz allerdings nur ca. 9 Tage.

## A 2.4 Codezeile

**A 1.4.1** Bestimmen Sie messtechnisch die Abarbeitungszeit von `P5OUT ^= 0x10;` bei Nutzung der `XT2CLK` und der `LFXT1CLK` Taktquelle.

Bei diesem Programm wird das Ausgangssignal periodisch invertiert, sodass der Signalwechsel via Messgerät erfassbar wird:

```

1 // Output des Takts
2 P5SEL |= ~(1 << 4);
3 P5DIR |= (1 << 4);
4 // Taktquelle einstellen
5 BCSCCTL2 = (BCSCCTL2 & ~SELM_3) | SELM_3; // LFXTCLK
6 BCSCCTL2 = (BCSCCTL2 & ~DIVM_3) | DIVM_0; // Divisor 1
7 // Signal invertieren
8 while(1)
9 {
10     P5OUT ^= 0x10;
11 }
```

Dieses Programm löst die Aufgabe für die `LFXT1CLK`-Clock; das Programm für `XT2CLK` ist analog - hier wird nur eine andere Taktquelle ausgewählt.

Dann ergibt sich für die Messungen:

**LFXT1CLK** Gemessene Frequenz  $f$  am Port: ca. 2,34 kHz

Die Periodenlänge  $T$ , die die Länge eines  $0 \rightarrow 1 \rightarrow 0$ -Durchlaufs angibt, ergibt sich via Formel  $T = \frac{1}{f}$ , also ist

$$T = \frac{1}{f} = \frac{1}{2,34 \text{ kHz}} = 427,4 \mu\text{s}$$

. Die Abarbeitung des Befehls entspricht genau den halben Periodenlänge der Schwingung. Also dauert die Abarbeitung des Befehls ungefähr  $213,7 \mu\text{s}$ .

**XT2CLK** Gemessene Frequenz  $f$  am Port: ca. 526,65 kHz

Die Abarbeitungszeit  $D$  ergibt sich analog.

$$D = \frac{1}{2}T = \frac{1}{2f} = \frac{1}{2 \cdot 526,65 \text{ kHz}} = 949,4 \text{ ns}$$

Also dauert die Abarbeitung des Befehls ungefähr  $949,4 \text{ ns}$ .