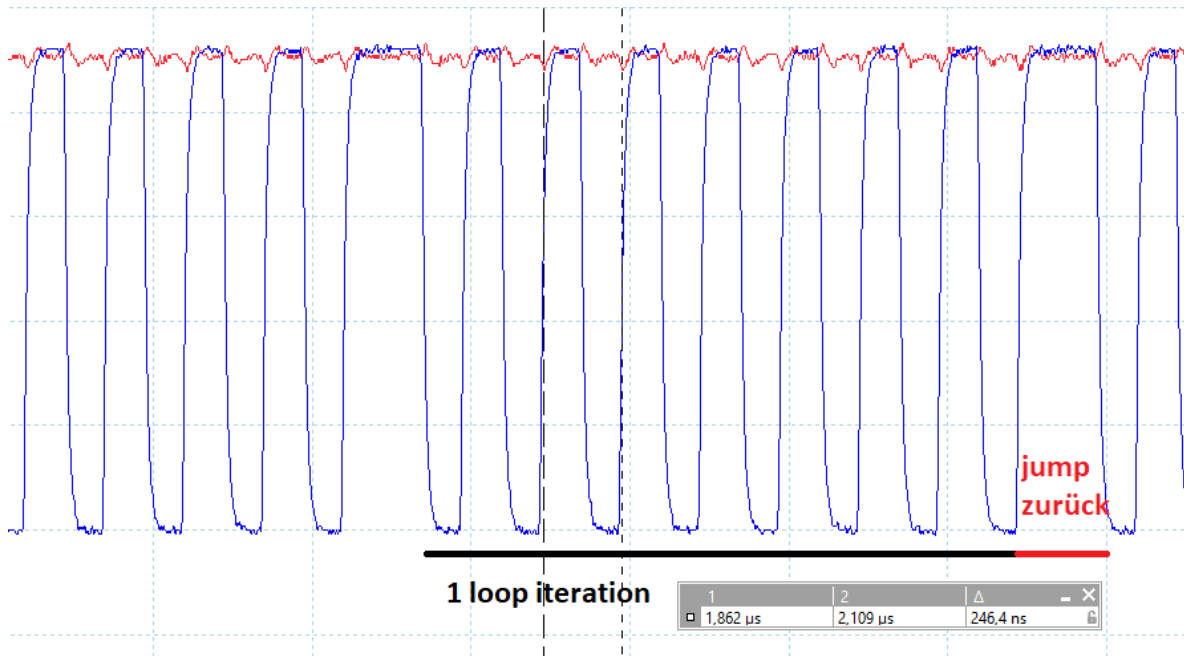


Loop Unrolling Effekt

Ursprünglich symmetrische Clock Periode von ca. **631ns** und daher **1,584MHz**.

Nun mit loop unrolling (hier 16-fach):



Das Toggeln dauert jetzt für 8 Perioden also nur noch ca. **250ns** und daher werden maximal circa **4MHz** erreicht. Wegen dem „jump zurück“ dauert eine Periode natürlich ein bisschen länger. Je mehr Loop unrolling man macht desto näher kommt man den 4MHz.

Die durchschnittliche Frequenz abhängig vom Loop Unrolling:

$$f(u) = \frac{1}{\frac{t_{\text{schnell}} * (\frac{u}{2} - 1) + t_{\text{jump}}}{\frac{u}{2}}} = \frac{u}{2 * (t_{\text{schnell}} * (\frac{u}{2} - 1) + t_{\text{langsam}})}$$

wo u der unroll-faktor ist und $t_{\text{schnell}} = 250 * 10^{-9}s$ und $t_{\text{langsam}} = 375 * 10^{-9}s$.

t_{langsam} ist die Periode mit dem langen High-Pegel (ausgelöst von dem `rxjmp` zum Loop-Anfang) und der erste Toggle danach.

Ein paar Werte für u :

$u=8$: 3,55MHz

$u=16$: 3,76MHz

$u=32$: 3,88MHz

$u=64$: 3,94MHz

...

Ich habe jetzt mal $u=16$ genommen für den Code. Das ist ja schon mal deutlich(!) besser als die 1,5MHz von zuvor... 😊

Bitbang Protokoll

Bei dem Bitbang Protokoll in dem Fabulous Wiki stand nichts über die Bit-Reihenfolge. Weil da nichts explizit steht gehe ich davon aus, dass man das MSB zuerst übertragen muss. Bei der Byte Order ist es dann vermutlich auch so, dass man das 0xFA vor dem 0xB0 Byte senden muss?

In dem Bild sieht man eine Übertragung von dem Control-Byte 0xA8 und dem Daten-Byte 0xA5. Sieht das für dich so korrekt aus?

