# Inbetriebnahme FMAC

In diesem Kapitel wird die FMAC-Einheit (Filter Math Accelerator) konfiguriert und in Betrieb genommen. Um eine korrekte Funktion nachzuweisen werden verschiedene Testfälle definiert und ausgeführt.

## Simple Moving Average Filter

Um eine korrekte Funktion des FMAC FIR-Filters nachzuweisen wird dieser zuerst ein einfaches gleitender Mittelwertbilder konfiguriert, welcher in englischsprachiger Literatur unter dem Begriff SMA (Simple Moving Average) bekannt ist.

Bei dieser Filterklasse entspricht die Ausgabe dem Durchschnitt der Eingangswerte über ein endliches Fenster. Mathematisch lässt sich ein diskreter Mittelwertbilder wie folgt beschreiben (Formel X.X):

– das Ausgangssignal am Zeitpunkt n

– das Eingangssignal am Zeitpunkt n

– Anzahl der Datenpunkte nach dem Mittelwert

– Anzahl der Datenpunkte vor dem Mittelwert

Frequenzgang:

(Quelle: Oppenheim A. V., & Willsky A. S. with Hamid N. (1996). Signals and systems (2nd ed.). Prentice Hall Inc.. ISBN 0-13–814757–4, S. 476)

Da bei einem FIR-Filter sich um ein kausales System handelt, hängt der Ausganswert y[n] nur von dem momentanen Wert x[n] und den vergangenen Werten x[n-1], x[n-2].. ab. Deshalb gilt für einen SMA FIR-Filter folgendes:

Die Ausgangssequenz eines LTD-Systems (linear, zeitinvariant und diskret) entsteht als Faltung von Eingangssignal und Impulsantwort.

Die Impulsantwort h[n] eines FIR-Systems entspricht exakt der Folge der Koeffizienten {b\_i} in der Differenzengleichung, die das System beschreibt.

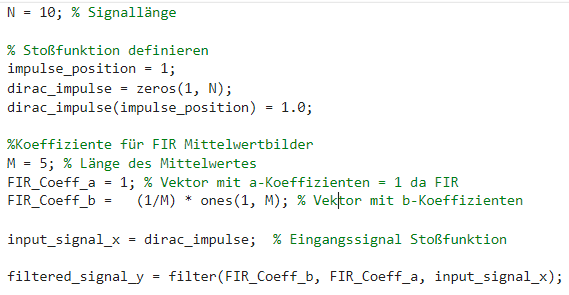
(Quelle : Meyer, Martin. Signalverarbeitung: .( 9. Aufl.), Springer Vieweg 2020. ISBN 978-3-658-32801-6, s. 234).

Zum Testen der korrekten Funktion der FMAC-Einheit wird in zuerst ein Matlab-Modell erstellt.

Es wird ein SMA-Filter mit 5 Taps verwendet.

Auf den Eingang des Filters wird ein Dirac-Stoß gegeben. Der Impulsantwort ist entsprechend der Definition eine Folge der Koeffizienten des Filters. Daraus folgt dass Impulsantwort eines SMA-Filters ein Rechteckimpuls ist. Für einen SMA-Filter mit N = 5:

{ x[0] + x[1] + .. x[4]}



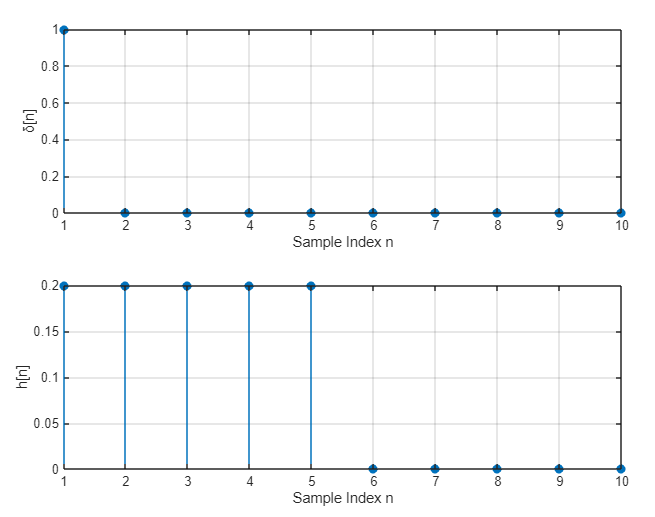


Abbildung 1: Impulsantwort eines SMA-Filters mit 5 Taps

## Datenformat Q15

In der digitalen Signalverarbeitung wird oft mit Festkomma-Darstellung gearbeitet.

Beispielweise wird in der FMAC Einheit Q15 Format verwendet.

**Festkomma-Darstellung** stellt Zahlen mit einem ganzzahligen und einem gebrochenen Teil in Zweierkomplement-Format dar.

Die Darstellung einer Q15-Zahl ist:

Vorzeichenbit (das 15. Bit) .

die Ziffer für n-te Bit.

Der Zahlenbereich, der in einer Q15-Zahl unterstützt wird, liegt zwischen -1,0 und 1,0, entsprechend der kleinsten und größten darstellbaren Ganzzahl, -32768 bzw. 32767. Beispielsweise wird die Zahl 0,25 in Q15 als 0x2000 (8192) codiert.

The filter coefficients are most likely calculated in floating point format and must be converted to 16-bit fixed point.

The format used by the FMAC is q1.15, which means that bits 0 to 14 represent fractional digits (ie. to the right of

the binary point), while bit 15 represents the sign/integer. Q1.15 format can therefore represent numbers in the

range -1.0 (0x8000) to +0.999969482421875 (0x7FFF).

If it is properly designed, the adaptive algorithm should only generate coefficient absolute values less than or

equal to 1, so all coefficients should be within the q1.15 numeric range. If this is not the case, then all coefficients

must be divided by the largest coefficient value, or greater.

Bei der Durchführung von Operationen mit Festkommazahlen lautet die Gleichung:

(Quelle: an4841 Digital signal processing for STM32 microcontrollers using CMSIS)

Im Laufe der Inbetriebnahme wurde zuerst folgende empfohlene Konvertierung von float in Q15-Format angewendet.

aFilterCoeffB\_q15[i] = (int16\_t)(aFilterCoeffB[i]\*0x8000); // Convert from float to Q1.15 format

aOutputValues[i] = (**float**)aOutputValues\_q15[i]/(**float**)0x8000; // convert from Q1.15 to float

(Quelle: an5305 Digital filter implementation with the FMAC using STM32CubeG4 MCU Package)

Da die Berechnung in Matlab in Double-Format erfolgt und FMAC-Einheit mit Q15-Format rechnet, folgt eine Konvertierung zur Genauigkeitsverlust. Dies führte dazu, dass die Ergebnisse aus dem Matlab-Modell und dem FMAC nicht identisch waren.

Aus diesem Grund wird beschlossen ein Matlab-Modell in Q15-Format aufzustellen welches ist eine Interpretation einer 16-Bit Integer Zahl ist.

Das Ergebnis einer Simulation vom Impulsantwort vom SMA FIR-Filter mit ist in der Abbildung 2 zu sehen.

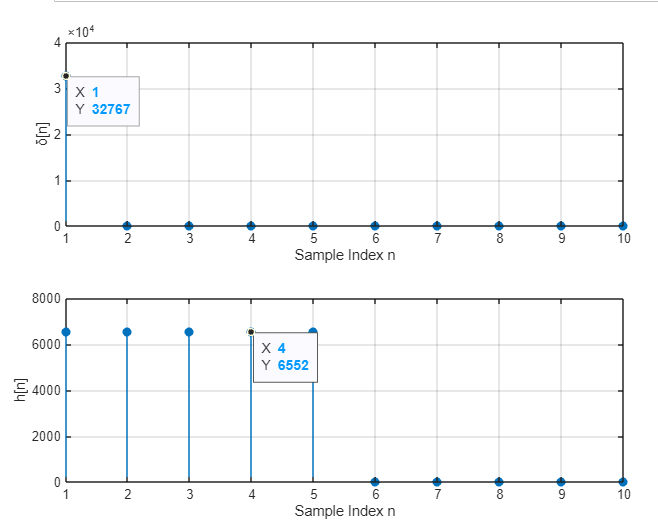


Abbildung 2: Impulsantwort eines SMA-Filters mit 5 Taps in Q15-Format (16-Bit Integer)

## Testfall 1: FMAC FIR SMA-5 Filter im Poling-Mode

Der FIR-Filter soll das Eingangssignal mit dem SMA der Fensterlänge 5 falten. Da alle Koeffizienten gleich sind, entspricht der SMA an jeder Stelle einfach dem Durchschnitt der letzten 5 Eingangswerte. Im stationären Zustand (ab der fünften Stelle des Ausgangssignals) soll der Filterausgang konstant den Wert 0 liefern, da alle Eingangswerte null sind.

Testvektoren:

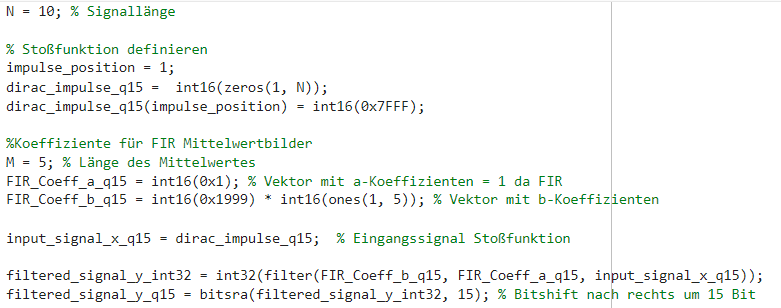
(int16) b = [6553, 6553, 6553, 6553, 6553]

(int16) x = [32767, 0, 0, 0, 0, 0]

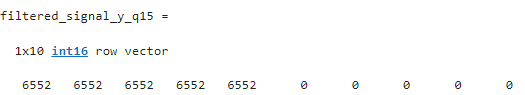
Erwarteter Ausgang:

(int16) y = [6552, 6552, 6552, 6552, 6552, 0, 0, 0, 0, 0]

Matlab-Modell:



Ausgabe in Matlab:



### Verifikation des Matlab-Modells

Der Ausgangswert 𝑦[𝑛] wird durch die Summe der Produkte der Koeffizienten und der entsprechenden Eingabewerte berechnet.

Da die Multiplikation zweier int16-Zahlen einen int32-Wert ergibt, muss dieser um 15 Bit nach rechts geschoben werden. Diese entspricht einer Division durch 32768:

Das Ergebnis aus dem Matlab-Modell stimmt mit der Berechnung überein. Das Modell ist nun verifiziert.

### Testausführung

Die FMAC-Einheit wird mithilfe der zur Verfügung stehenden HAL-Funktionen (Hardware Abstraction Layer) und Strukturen wie folgt konfiguriert:

* Länge X1-Buffer (Eingangsbuffer): 5
* Länge X2 (B-Koefizienten-Buffer): 5
* Länge A-Koefizienten-Buffer: 0 (FIR -Filter)
* Länge Y-Buffer (Augangsbuffer): 5
* X1-Preload: 5 Samples mit „0“
* Watermark X1-Buffer: 0
* Watermark Y1-Buffer: 0
* Funktion: Convolution (Faltung)
* X1-Buffer Zugriff: Polling
* Y-Buffer Zugriff: Polling
* Clipping Disabled (muss noch untersucht werden)
* Filter Gain 1 (keine Verstärkung)

Alle Konfigurationsparameter werden der Struktur sFmacConfig zugewiesen. Diese wird samt hfmac Handle an die Funktion HAL\_FMAC\_FilterConfig() übergeben.

Als nächstes muss der X1-Buffer mit 5 Nullwerten vorgeladen werden. Dies geschieht mit der Funktion HAL\_FMAC\_FilterPreload()

Der Filter kann nun mit HAL\_FMAC\_FilterStart() gestartet werden.

Nach dem Start werden in jeder Iteration einer For-Schleife jeweils ein Wert geschrieben und ein Wert gelesen. Vor jedem Schreiben des WDATA-Registers wird „gepolt“ bis der FMAC\_SR\_X1FULL nicht gesetzt ist. Analog wird vor jedem Lese-Zugriff auf RDATA-Register gewartet bis FMAC\_SR\_YEMPTY nicht gesetzt ist (Y-Buffer ist nicht leer).

(Quelle RM0440)

Schließlich werden die Eingangs- und die Ausgangswerte mit der **printf**-Funktion in der Konsole ausgegeben Abbildung 3. Dazu wurde USART2 konfiguriert.

Der vollständige Sourcecode: <https://github.com/evgenmlnk/teamproject_stm32/tree/main/FMAC/nucleo-g474re-fmac-moving-average-test1>

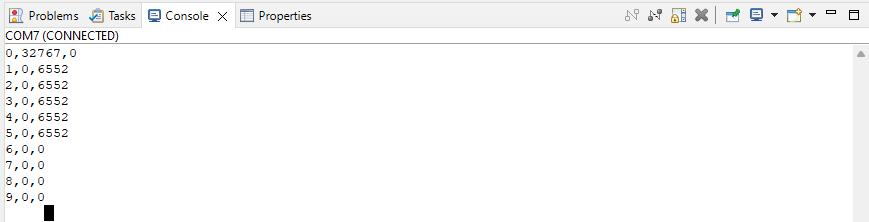


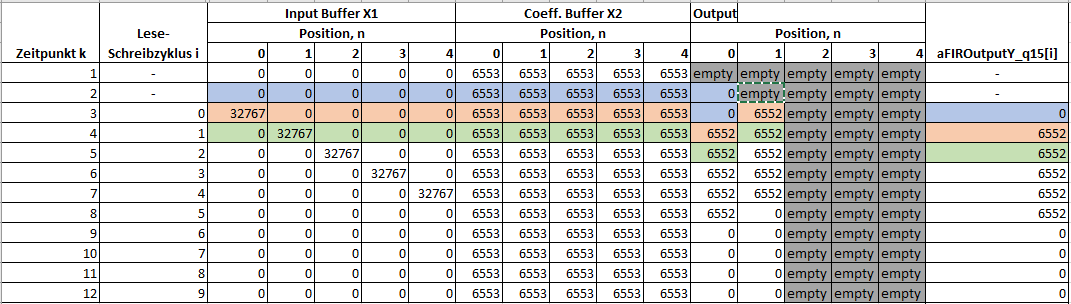
Abbildung 3: Konsole-Ausgabe FMAC FIR SMA-5 (Lese-Schreibzyklus, Eingang X, Ausgang Y)

## Bewertung der Ergebnisse

Die Ausgangswerte Y[n] sind zwar vom Betrag wie erwartet, allerdings ist der Ausgang um ein Sample verzögert.

An dieser Stelle wird die Arbeit der FMAC-Einheit mi dem Fokus auf die Circular Buffers X1 und Y analysiert. Die Tabelle 1 soll dazu dienen die Einzelne Zustände der Buffers zu veranschaulichen (vollständige Tabelle siehe Beigefügte Excel-Datei).

Tabelle 1

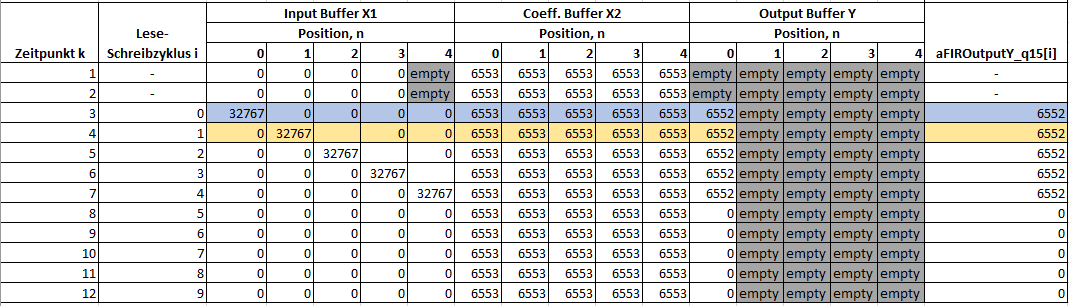


Zum Zeitpunkt k=1 wird der X1 Buffer vorgeladen. Zum Zeitpunkt k=2 wird die FMAC-Einheit gestartet. Da der X1 Buffer bereits vorgeladen wurde ist die Anzahl der Werte bereits genug für die Kalkulation. Deshalb wird in diesem Zyklus eine Berechnung durchgeführt und das Ergebnis in den Output Buffer auf die Position 0 geschrieben. Der Wert x[0] wird nach der Berechnung „discarded“.

Zum Zeitpunkt k=2 wird der neue Wert in den X1 Buffer geschrieben. Der neu berechnende Wert Y[1] wird in die Position 1 vom Y-Buffer geschrieben. Der Wert Y[0] wird ausgelesen, usw.

Um eine Verzögerung zu vermeiden kann der X1-Buffer mit einen Werte weniger vorgeladen werden wie in der Tabelle 2 zu verdeutlichen ist.

Tabelle 2



Der PRELOAD\_SIZE wird nun von 5 auf 4 reduziert (Abbildung 4).

Ausgabe (Lese-Schreibzyklus, Eingang X, Ausgang Y) :

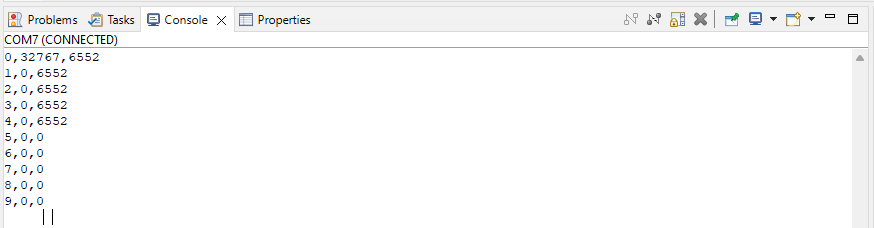
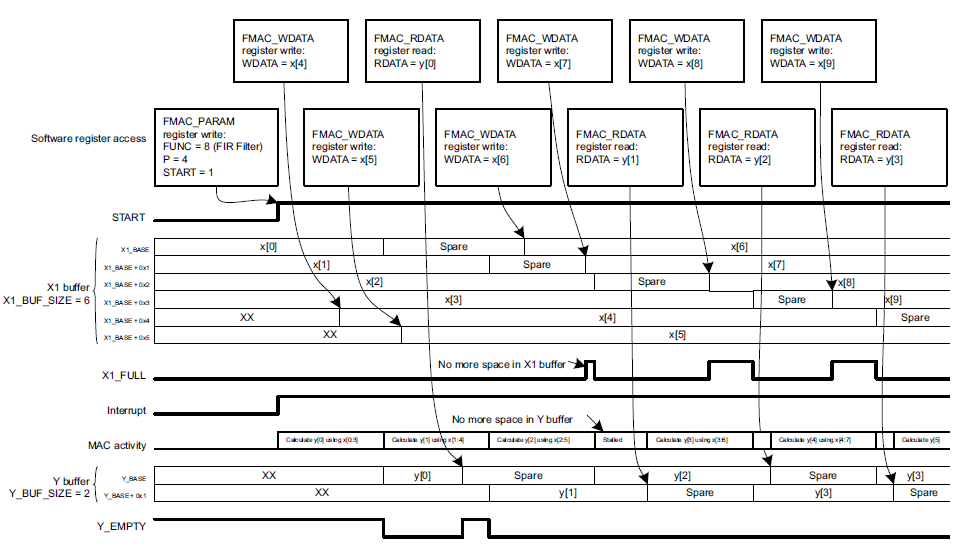


Abbildung 4: Konsole-Ausgabe FMAC FIR SMA-5 (Lese-Schreibzyklus, Eingang X, Ausgang Y)

### Testfall 2: FMAC FIR Interrupt-Mode



XX

XX

x[0]

x[1]

x[2]

x[3]

XX

XX

XX

x[1]

x[2]

x[3]

XX

XX

x[4]

x[1]

x[2]

x[3]

XX

XX

XX

x[4]

x[2]

x[3]

XX

x[7]

x[6]

x[5]

x[4]

x[3]

x[7]

x[6]

x[5]

x[4]

x[9]

x[8]

x[7]

x[6]

x[5]

XX

XX

XX

y[0]

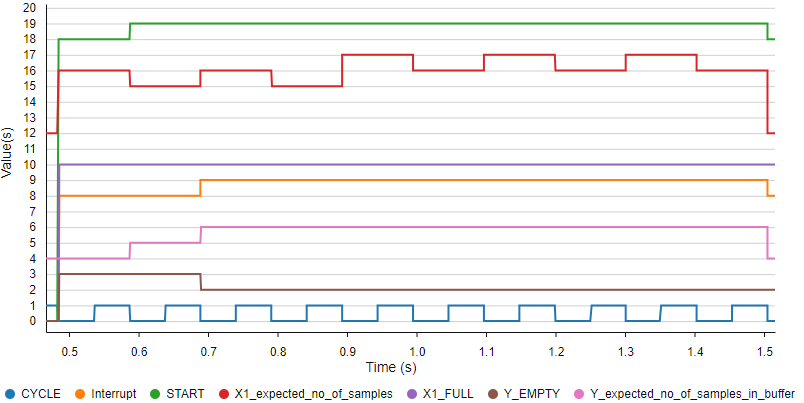
y[1]

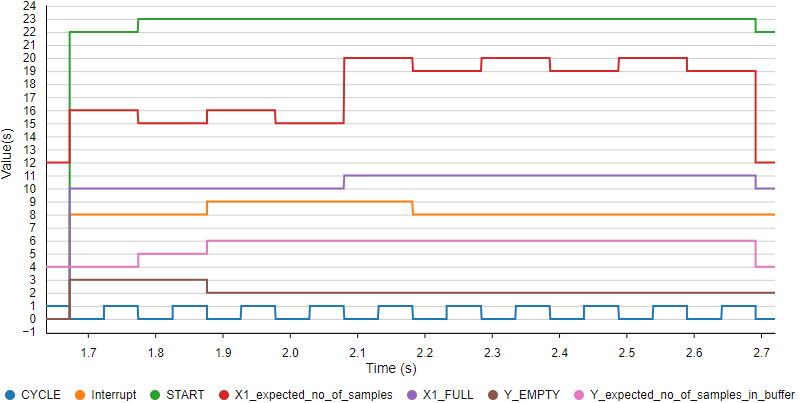
y[0]

X1-Buffer

Y- Buffer

Start





# FIR-Filter mit CMSIS-Library