

Berechnung des Speicheraufwands einer segmentierten Faltung

1. Variablenzuordnung

$x[n]$	Eingangssignal
L_x	Länge von $x[n]$
N	Länge eines Segments $x_k[n]$
$h[n]$	Impulsantwort
L_h	Länge von $h[n]$
P	Länge eines Segments $h_k[n]$
$y[n]$	Ausgangssignal
L_y	Länge von $y[n]$
S	Wortbreite der Anwendung in <i>Bit</i>

2. Betrachtung der Impulsantwort

Die Impulsantwort $h[n]$ wird zunächst in den Speicher geladen, sie braucht daher genau

$$L_h \cdot S \text{ Bit}$$

des Speichers. Dieser Zustand wird solange gehalten, bis der Transformationsvorgang für die Impulsantwort mit der Segmentierung beginnt. Dabei muss beachtet werden, dass ein Zero-Padding erforderlich ist, damit die Ergebnislänge nach der IFFT korrekt ist. Somit braucht die FFT genau jene Anzahl an Werten, die der Länge des Faltungsergebnisses entspricht. Somit benötigt ein Segment der Impulsantwort nach der FFT

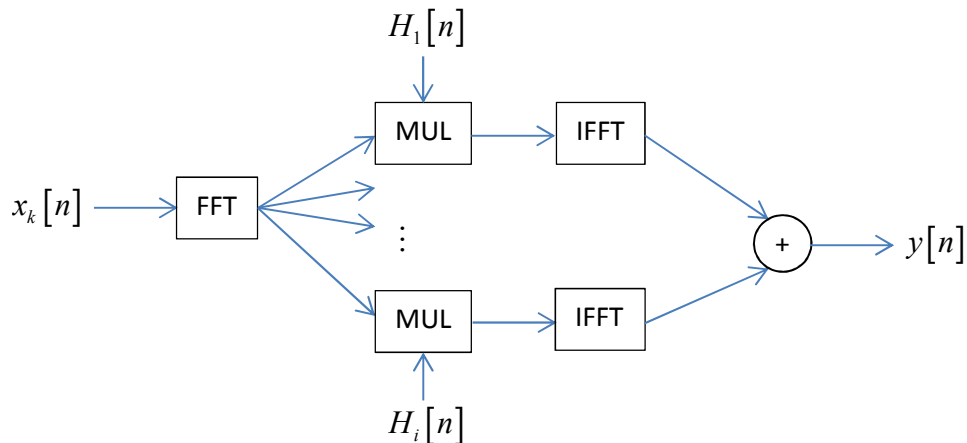
$$(2P - 1) \cdot S \text{ Bit}.$$

Die gesamte segmentierte Impulsantwort $H_i[n]$ braucht folglich

$$(2P - 1) \cdot \frac{L_h}{P} \cdot S \text{ Bit}$$

im Speicher. Dieser Speicher wird während der gesamten Berechnung immer wieder benötigt, er darf auch nicht überschrieben werden.

3. Betrachtung des Ablaufs für ein Segment



Läuft ein Segment $x_k[n]$ in den Prozessor, wird es zunächst transformiert. Dies kann „In-Place“ passieren, sodass kein zusätzlicher Speicher erforderlich ist. Die erforderliche Speichergröße für ein transformiertes Segment von $x[n]$ ergibt sich aus den für die Impulsantwort bereits gemachten Betrachtungen. Sie beträgt $(2N-1) \cdot S \text{ Bit}$.

Anschließend wird das transformierte Segment $X_k[n]$ mit allen Segmenten von $H[n]$ multipliziert. Die Größe der Ergebnisse dieser Multiplikationen können anhand von Vorbetrachtungen abgeschätzt werden. Multipliziert man im Binärbereich, so ist das Ergebnis genau so lang wie die Summe der Länge der Faktoren. Ein Beispiel mit 4 Bit:

$$\begin{array}{r}
 \overbrace{1111}^{\text{Länge } N} \\
 \cdot \quad 1111 \\
 \hline
 \underbrace{1110001}_{\text{Länge } N+N}
 \end{array}$$

Eine Multiplikation zweier transformierter Segmente von $x[n]$ und $h[n]$ hat demnach die Länge

$$(2 \cdot (N-1)) + (2 \cdot (P-1)) \cdot S \text{ Bit} = 2 \cdot (N+P-2) \cdot S \text{ Bit}$$

Diese Multiplikation muss so oft durchgeführt werden, wie es Segmente von $H[n]$ gibt, also

genau $\frac{L_h}{P}$ -mal. Somit benötigen die Ergebnisse der Multiplikationen

$$\frac{2 \cdot L_h}{P} \cdot (N+P-2) \cdot S \text{ Bit}$$

Anschließend liegen die Ergebnisblöcke $Y_j[n]$ vor, welche nun durch die IFFT laufen. Diese kann wieder „In-Place“ durchgeführt werden, somit wird es an dieser Stelle keinen zusätzlichen Speicherbedarf geben.

Das sich anschließende Overlap-Add-Verfahren kann durch ein Halten des Ergebnisses in einem der Register der ALU ohne zusätzlichen Speicherbedarf durchgeführt werden. Es könnte allerdings nötig sein, die Ergebnisse der Summen zwischen zu speichern, deswegen wurde auch an dieser Stelle ein Ansatz entwickelt. An einer Summe sind maximal $2 \cdot \frac{L_h}{P}$ Summanden beteiligt. Diese haben die Wortbreite S , weil es einzelne Samples aus einem Segment $y_j[n]$ sind. Somit wäre für einen Speicher die binäre Größe von $2 \cdot \frac{L_h}{P}$ addiert mit der Wortbreite S nötig. Der Wert $2 \cdot \frac{L_h}{P}$ kann maximal

$$2 \cdot \frac{88200}{2} = 88200 \Rightarrow [10101100010001000]_2 \hat{=} 17 \text{ Bit}$$

betragen. Je nachdem, welcher Wert für S gewählt wird, ergibt sich entsprechend zusätzlicher Speicherbedarf. Dieser kann allerdings in einer Abschätzung vernachlässigt werden, weil er sehr klein im Gegensatz zum Gesamtablauf ist.

4. Ermittlung des Speicherbedarfs

Aus der Einzelbetrachtung kann nun eine Prognose gemacht werden, wie der Gesamtspeicheraufwand sein wird. Dazu betrachtet wird jener Bereich der Rechnung, indem die maximale Überlagerungsanzahl erreicht wird. Dies ist genau der Bereich der Berechnung, der die größte Speicherauslastung aufweisen wird. Auf Grund des Overlap-Add-Verfahrens müssen zu dieser Zeit doppelt so viele Segmente von $x[n]$ verarbeitet werden, wie es Segmente der Impulsantwort $h[n]$ gibt. Demnach sind $2 \cdot \frac{L_h}{P}$ Segmente in der Berechnung aktiv und müssen mit den bereits ermittelten Speicheranforderungen betrachtet werden. Ebenso muss die Betrachtung noch um einen Schritt weiter gemacht werden, denn ein Segment $y_j[n]$ und der dafür benötigte Speicher muss genau so lange gehalten werden, bis das übernächste Segment eintrifft. Daher sind zur Speicherbetrachtung $3 \cdot \frac{L_h}{P}$ Segmente nötig. Dieser ist demnach:

$$\underbrace{\left(\frac{(2P-1) \cdot L_h}{P} \right)}_{H_l[n]} + \left[\underbrace{(2N-1)}_{\text{Segment } X_k[n]} \cdot \underbrace{\frac{L_h}{P}}_{\text{neues Laden von Segmenten}} \right] + \left[\underbrace{(P+N-2)}_{\text{Segment } Y_j[n]} \cdot \underbrace{\frac{3 \cdot L_h}{P}}_{\text{max. Überlagerung + Laden der Folge-Blöcke}} \right]$$

Wertetabellen:

a. Speicherbedarf für $H_i[n]$ und Gesamtbedarf

Lx = Lh = Länge h[n] & x[n]			88200
N	P	Größe $H_i[n]$	Speicherbedarf (in Samples)
1	1	0	0,00E+00
2	2	1,32E+05	5,29E+05
4	4	1,54E+05	7,06E+05
8	8	1,65E+05	7,94E+05
16	16	1,71E+05	8,38E+05
32	32	1,74E+05	8,60E+05
64	64	1,75E+05	8,71E+05
128	128	1,76E+05	8,76E+05
256	256	1,76E+05	8,79E+05
512	512	1,76E+05	8,81E+05
1024	1024	1,76E+05	8,81E+05
2048	2048	1,76E+05	8,82E+05
4096	4096	1,76E+05	8,82E+05
88200	88200	1,76E+05	8,82E+05

b. Speicherbedarf für verschiedene Wortbreiten S

Lx = Lh = Länge h[n] & x[n]			88200			
N	P	S = 8 bit	S = 16 bit	S = 24 Bit	S = 64 bit	
1	1	0,00E+00	0,00E+00	0,00E+00	0,00E+00	
2	2	4,23E+06	8,47E+06	1,27E+07	3,39E+07	
4	4	5,64E+06	1,13E+07	1,69E+07	4,52E+07	
8	8	6,35E+06	1,27E+07	1,91E+07	5,08E+07	
16	16	6,70E+06	1,34E+07	2,01E+07	5,36E+07	
32	32	6,88E+06	1,38E+07	2,06E+07	5,50E+07	
64	64	6,97E+06	1,39E+07	2,09E+07	5,57E+07	
128	128	7,01E+06	1,40E+07	2,10E+07	5,61E+07	
256	256	7,03E+06	1,41E+07	2,11E+07	5,63E+07	
512	512	7,04E+06	1,41E+07	2,11E+07	5,64E+07	
1024	1024	7,05E+06	1,41E+07	2,12E+07	5,64E+07	
2048	2048	7,05E+06	1,41E+07	2,12E+07	5,64E+07	
4096	4096	7,05E+06	1,41E+07	2,12E+07	5,64E+07	
88200	88200	7,06E+06	1,41E+07	2,12E+07	5,64E+07	