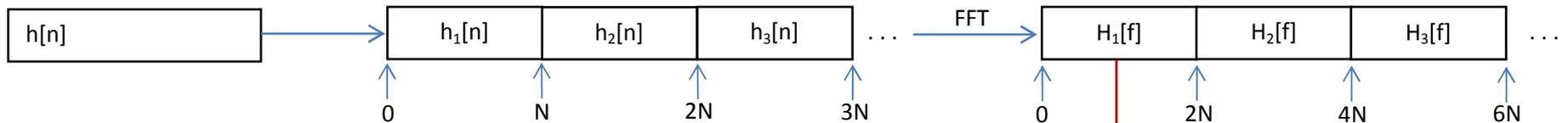


Segmentierte Faltung – Analyse des Gesamtprozesses zur Übersetzung in Programmstruktur

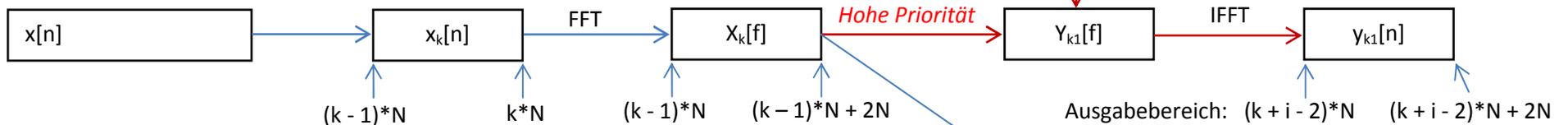
globale Variablen (→ *es gibt bestimmt noch mehr, die eingeführt werden müssen*):

Segmentlänge $N (= P)$ → int N
 Länge h → int len_h
 Anzahl Segmente h → int $segnum_h = len_h / N$
 Index für Eingangssegmente → int $xindex$
 Initialisierung Eingangskanal → float x

Impulsantwort transformieren und speichern:

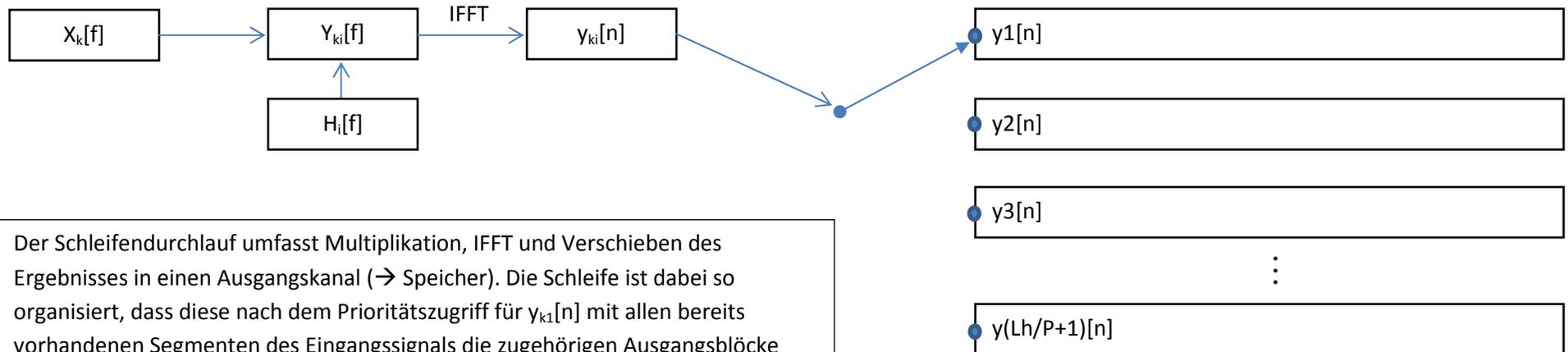


Beliebiges Eingangssegment bearbeiten:



Die erste Faltung von $x_k[n]$ mit $h_1[n]$ wird direkt im nächsten Schritt benötigt und sollte daher übergeordnet zur Dauerschleife sein. In dieser werden alle weiteren Ergebnis-Segmente berechnet.

Programm-Dauerschleife zur Berechnung der Ergebnissegmente:



Der Schleifendurchlauf umfasst Multiplikation, IFFT und Verschieben des Ergebnisses in einen Ausgangskanal (→ Speicher). Die Schleife ist dabei so organisiert, dass diese nach dem Prioritätszugriff für $y_{k1}[n]$ mit allen bereits vorhandenen Segmenten des Eingangssignals die zugehörigen Ausgangsblöcke berechnet. Dabei wird immer das jeweils neuste Element des Eingangssignals zuerst betrachtet. Die Schleife durchläuft dann die weiteren Segmente bis zur Berechnung des Ergebnisses aus dem ältesten verfügbaren Eingangssegment mit dem aktuell höchsten Segment der Impulsantwort. Die Schleife weiß durch einen Übergabewert, wie viele Segmente $X_k[f]$ aktuell vorliegen und wird entsprechend erweitert.

Ausgangskanäle: Hier werden die Segmente $y_{ki}[n]$ nach der Berechnung abgelegt. Die Overlap-Add Operation erhält aus diesen Kanälen die Werte zur Berechnung des Ausgangssignals.

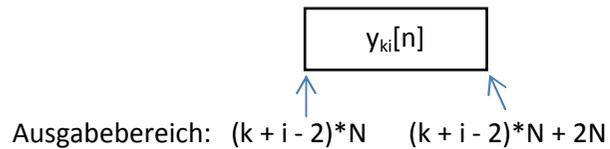
[Amn.: Hier lässt sich noch optimieren, um die Anzahl der Kanäle zu verringern. → Einer der nächsten Arbeitsschritte]

Schleifendurchlauf im Detail:

Die erste Berechnung ist y_{12} . y_{11} wird durch die Prioritäts-Routine erledigt. Die Schleife hat zu diesem Zeitpunkt den Wert 1 erhalten, weil es nur ein Eingangssegment gibt. Sie würde daher weiter die Ergebnisse y_{1i} berechnen, bis die Prioritäts-Routine das Segment $x_2[n]$ holt und verarbeitet. Ab diesem Zeitpunkt hat die Schleife nun den Wert 2 und berechnet nun ein Ergebnis y_{2i} , dann ein Ergebnis y_{1i} und beginnt wieder von vorne. Durch diesen Ablauf kann die Zeit bis zur maximalen Überlagerung genutzt werden, um so viele Ergebnissegmente wie möglich zu berechnen, sodass Berechnungsergebnisse bereits vorliegen und der Prozess dadurch beschleunigt werden kann. Durch den Schleifenablauf ist zudem gewährleistet, dass ein Ergebnissegment für die nächste Ausgabe mit erhöhter Priorität behandelt wird als jenes für eine spätere Ausgabe.

Ausgabebereiche:

Der Ausgabebereich für ein beliebiges Ausgabesegment $y_{ki}[n]$ kann anhand des Indexes ermittelt werden. Dazu wird der Index von Eingangssegment und Impulsantwortsegment addiert, anschließend wird zwei subtrahiert. Jener Wert multipliziert mit der Segmentlänge N ist der Startwert in der Ausgabe, wenn man davon ausgeht, dass der erste Wert des Ausgangssignals $y_{11}[0]$ zum Zeitpunkt $t_0 = 0$ ausgegeben wird.



Somit ist es möglich, dass das Overlap-Add Verfahren mit diesen Bereichen arbeitet und entsprechend den aktuellen Wert berechnet.

Overlap-Add-Berechnung:

