Im ersten Schritt wird ein Dateiname einer einzulesenden Datei per Konsoleneinngabe übergeben. Dieser Dateiname wird im Anschluss an die Klasse SumSignal übergeben.

In SumSignal wird nun eine Referenz auf eine Datei erzeugt, sodass diese im Anschluss ausgelesen werden kann. Die Datei ist leerzeichensepariert aufgebaut, sodass die Filtereigenschaft von fscanf ausgenutzt werden kann um die einzelnen Summenbits des Summenstreams zu splitten und diese anschließend in ein Array zu schreiben.

Dieses Array wird im späteren Verlauf benötigt um die einzelnen Signale der jeweiligen GPS Satelliten aufzuteilen.

Für die Berechnung der einzelnen Signale erzeugt jeder GPS Satellit eine individuelle Chipsequenz, die auf zwei Mutterfolgen basiert.

Daher werden zwei unterschiedliche Mutterfolgengeneratoren aufgebaut. Mutterfolge 1 entsteht indem man ein 10 Bit Schieberegister am niederwertigsten Bit und an Bit 8 (Anfang der Zählung 1) XOR verknüpft und dieses Ergebnis wiederum als höchstwertiges Bit in das Schieberegister einfügt.

Analog zu dieser Arbeitsweise wird die zweite Mutterfolge generiert. Hierbei werden jedoch statt dem ersten und achten Bit die Bits 2, 3, 5, 8 und 9 XOR verknüpft.

Konkret bedeutet dies für die erste Mutterfolge: man liest das niederwertigste Bit des aktuellen Registerstatus aus und macht ein XOR auf das achte Bit des aktuellen Registerstatus.

Logisch betrachtet bedeutet dies, man wendet Bitmasken auf die aktuelle Sequenz an. Im ersten Fall wird eine Und Verknüpfung der aktuellen Sequenz mit der Bitmaske (1 << 0) – Binär: 00 0000 0001 – angewendet um das kleinste Bit zu bekommen. Das achte Bit erhält man durch eine Und-Verknüpfung der aktuellen Registersequenz mit der Bitmaske (1 << 7) – Binär: 1000 0000 - . Beide werden nun mit XOR verknüpft, sodass sich der logische Ausdruck (sequenz & (1 << 0) ^sequenz & (1 <<7) >> 7) ergibt.

Um nun die Sequenz zu shiften und den neuen Wert einzufügen wird ein Rechtsshift angewendet, sodass beispielsweise aus 1000 0000 – 0100 0000 wird und setzt mit Hilfe einer Oder-Verknüpfung das höchstwertigste Bit. Logisch bedeutet dies: sequenz = sequenz | (neuesBit << 9)

Die Chipsequenzen der jeweiligen Satelliten werden erzeugt indem man nun das niederwertigste Bit des aktuellen Schieberegisterzustands der ersten Mutterfolge mit einer Satellitenspezifischen Kombination von Verknüpfungen der zweiten Mutterfolge XOR verknüpft. Für Satellit 1 wird die XOR Verknüpfung von Bit 5 und 9 mit dem niederwertigsten Bit der ersten Mutterfolge XOR verknüpft.

Konstanten:.

BIT\_1 (1 << 0)

Niederwertigstes Bit des Schieberegisters (der Mutterfolge)

BIT\_2 (1 << 1)

Zweites Bit des Schieberegisters (der Mutterfolge)

BIT\_3 (1 << 2)

Drittes Bit des Schieberegisters (der Mutterfolge)

BIT\_5 (1 << 4)

Fünftes Bit des Schieberegisters (der Mutterfolge)

BIT\_8 (1 << 7)

Achtes Bit des Schieberegisters (der Mutterfolge)

BIT\_9 (1 << 8)

Neuntes Bit des Schieberegisters (der Mutterfolge)

HIGH\_BIT 9

Höchstwertiges Bit des Schieberegisters der Mutterfolge

INITIAL\_VALUE 1023

Initialwert zur Initialisierung der Schieberegister zur Generierung der Mutterfolgen. Festgesetzter Wert in diesem Verfahren (11 1111 1111)

REGISTER\_WIDTH 10

Anzahl der D-Flip-Flops des Schieberegisters zur Generierung der Mutterfolgen. Festgesetzter Wert in diesem Verfahren