

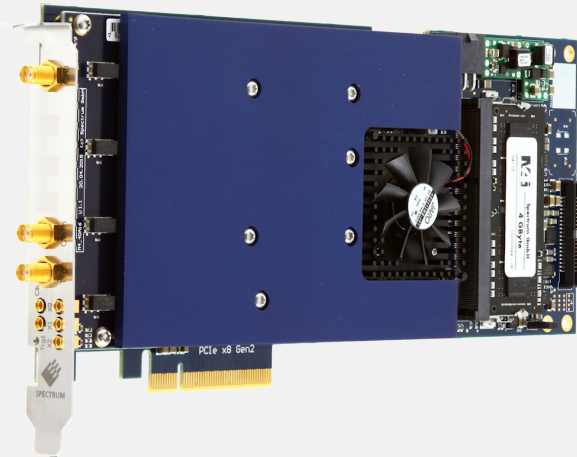
► White Paper

Software für schnelle, blockweise Mittelwertbildung

Hintergrund

Der auf Speicherblöcken bzw. segmentiertem Speicher basierende Modus für die Mittelwertbildung wird bei Digitizern für verschiedene Anwendungen eingesetzt, bei denen inkohärentes Rauschen aus einem Signal entfernt werden muss. Unabhängig vom Hersteller des Digitizers begrenzen sämtliche FPGA-basierten Hardware-Implementierungen des Modus für die blockweise Mittelwertbildung die maximale Größe des Segments, dessen Mittelwert bestimmt werden soll. Die Begrenzung hängt von der Kapazität des FPGA ab und liegt in der Regel bei 32.000 bis 500.000 Werten.

In diesem Whitepaper wird erläutert, wie man die schnellen PCIe-Streamingfähigkeiten der Digitizer der M4i Serie von Spectrum zur Implementierung einer blockweisen Mittelwertbildung in der Software nutzt, um diese Grenzen zu überwinden. Die Ergebnisse der hardware- und softwarebasierten Methoden für die blockweise Mittelwertbildung werden unter Berücksichtigung der mit dem Digitizer M4i.2230-x8 (1 Kanal, 5 GS/s, 8-Bit-Digitizer mit 1,5 GHz Bandbreite) erzielten Ergebnisse verglichen.

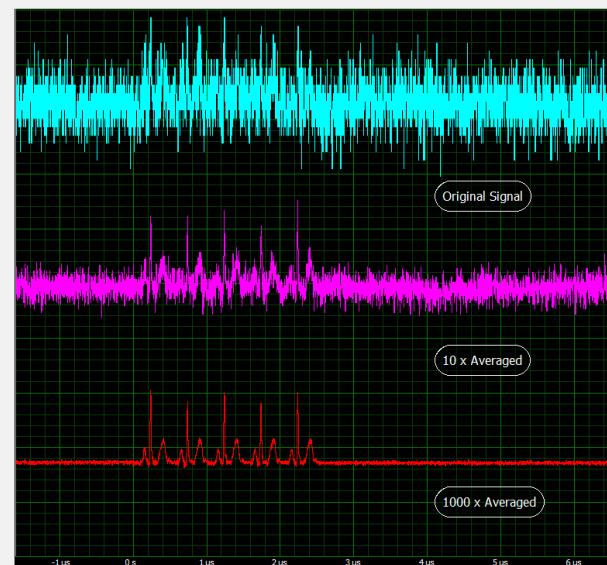


M4i.2230-x8: 5 GS/s Digitizer mit 1.5 GHz Bandbreite

Was versteht man unter der blockweisen Mittelwertbildung?

Der Modus „Block Averaging“ (blockweise Mittelwertbildung) kann zur Verbesserung der Signaltreue eines beliebigen periodischen Signals durch Entfernen seiner zufällig auftretenden Rauschteile verwendet werden. Der Modus ermöglicht es, mehrere einzelne Aufzeichnungen durchzuführen, zu akkumulieren und deren Mittelwert zu bilden. Der Prozess verringert den Anteil statischen Rauschens und verbessert die optische Darstellung des periodischen Signals. Das gemittelte Signal weist eine verbesserte Messauflösung und einen erhöhten Signal-Rausch-Abstand auf.

Der Modus „Block Averaging“ kann zur Verbesserung von Messungen in verschiedenen Anwendungen verwendet werden, beispielsweise bei Radartests, Ultraschallprüfungen, Glasfaserprüfungen, Laserentfernungsmessungen, in der Massenspektroskopie oder in der medizinischen Bildgebung.



Verbesserung des Signal-Rausch-Abstands durch die blockweise Mittelwertbildung

Im Screenshot rechts sind ein Low-Pegel-Signal (ca. 2 mV) zu sehen, das vollständig von

► White Paper

statischem Rauschen überlagert wird, sowie die Verbesserung, die sich durch Anpassung verschiedener Parameter für die Mittelwertbildung erzielen lässt. Während das Quellensignal in der ursprünglichen Einzelaufzeichnung überhaupt nicht zu sehen ist, erkennt man nach einer 10-maligen Mittelwertbildung ein Signal mit 5 Spitzen. Durch eine 1000-malige blockweise Mittelwertbildung lässt sich die Signalqualität nochmals verbessern und die tatsächliche Form des Signals einschließlich sekundärer maximaler und minimaler Spitzenwerte erkennen.

Bei diesem Beispiel wurde ein Digitizer mit einer Abtastrate von 500 MS/s (2 ns pro Punkt) und einer 14-Bit-Auflösung verwendet.

Systemeinrichtung

Das Testsystem bestand aus einem herkömmlichen Büro-PC aus der Entwicklungsabteilung von Spectrum und den folgenden Komponenten:

- Motherboard: Gigabyte GA-H77-D3H
- Prozessor: Intel i7-3770 4 x 3.4 GHz
- Speicher: 8 GByte DDR3 memory
- SSD: 120 GByte Samsung 840 EVO
- Betriebssystem: Windows 7 Professional 64 Bit
- Compiler: Visual Studio 2005 Standard Edition

Das Motherboard verfügt über einen freien PCIe x8 Gen2-Steckplatz, der für die Digitizer-Karte genutzt wird. Dieser Steckplatz ist für eine Nutzdatengröße (payload size) von 256 Bytes ausgelegt, wodurch die M4i-Karten von Spectrum eine maximale Streaming-Geschwindigkeit von ca. 3,4 GB/s erreichen können (ohne jegliche Datenverarbeitung).

Software-Implementierung

Die Testsoftware wurde komplett in C++ geschrieben und basiert auf Streaming-Beispielen von Spectrum. Die Testkarte empfing ein externes Triggersignal und zeichnete ein Segment Daten je Triggerereignis auf. Die Daten wurden im On-Board-Speicher der Karte gespeichert und über Scatter-Gather-DMA direkt in den PC-Speicher übertragen, wo sie für die blockweise Mittelwertbildung akkumuliert wurden. Verschiedene Setups und Methoden der Verbesserung wurden getestet, um herauszufinden, wie leistungsstark das System sein kann.

Der kurze Auszug aus dem Quellcode zeigt die Threaded-Version der grundsätzlichen Summenbildung mittels einer Schleife. Dies ist der entscheidende und die Geschwindigkeit bestimmende Teil der Software.

```
SPCM_THREAD_RETURN SPCM_THREAD_CALLTYPE pvAverageSegmentPart (void*
pvArguments)
{
    SPCM_AVERAGE_DATA* pstData = (SPCM_AVERAGE_DATA*) pvArguments;
    int32 i, j;
    int32 lStart;
    int32 lEnd;
    int32 lNumSegments = pstData->lNumSegments;
    int32* plAverageData = pstData->plAverageData;
    int8* pbyData = pstData->pbyData;

    while (pstData->bRunNotQuit)
    {
        spcm_vWaitEvent (&pstData->hStart);

        for (j=0; j<lNumSegments; j++)
        {
            lStart = j * pstData->lSegmentsize + pstData->lStartOffset;
            lEnd = pstData->lStartOffset + pstData->lAveragesize;

            for (i=lStart; i < lEnd; i++)
                plAverageData[i] += (int32) pbyData[i];
        }

        spcm_vSignalEvent (&pstData->hEnd);
    }

    return 0;
}
```

► White Paper

Nachstehend sind Informationen und Kommentare zu den verschiedenen Aspekten der Implementierung aufgeführt, die im Abschnitt „Ergebnisse“ zusammengestellt sind:

- **Segmentgröße (Segmentsize):** die Anzahl der Werte je Datenssegment, die nach Empfang eines Triggerereignisses aufgezeichnet werden.
- **Mittelwerte (Averages):** die Anzahl der Mittelwerte (Summierungen), die bestimmt werden, bis ein Segment gespeichert und der Prozess der Mittelwertbildung von neuem gestartet wird.
- **Benachrichtigungsgröße (Notifysize):** die Datenmenge, nach deren Empfang die PC-Hardware einen Interrupt generiert. Der Parameter Notifysize bestimmt die Geschwindigkeit der kompletten Mittelwertbildungsschleife. Ist der Parameter Notifysize größer als der Parameter Segmentsize werden mehrere Segmente je Interrupt summiert. Dadurch verringert sich der Overhead für die Thread-Kommunikation und das Interrupt-Handling.
- **Zwischenspeichergröße (Buffersize):** der gesamte Zielpuffer im Speicher für die DMA-Übertragung. In unserem Beispiel ist die Größe des Zwischenspeichers als das 16-Fache des Parameters Notifysize festgelegt.
- **Triggerrate:** die Wiederholungsrate des externen Signalgebers. In den Ergebnissen ist die maximal erzielte Triggerrate ohne Überlauf der Zwischenspeicher aufgeführt.
- **Threads:** Um die Geschwindigkeit des Summierungsprozesses zu erhöhen, wurde dieser Task parallelisiert, indem die Summierung in verschiedene Software-Threads aufgeteilt wird. Wenn in der Spalte „Threads“ der Wert Null steht, wird im Summierungsprozess kein Threading angewendet, sondern der Summierungsprozess läuft direkt inline in einer Schleife ab.
- **CPU-Auslastung (CPU Load):** Da der Prozess der Mittelwertbildung in der Software abläuft, muss die CPU bzw. müssen die CPUs die gesamte Rechenleistung erbringen. Glücklicherweise bestehen heutige CPUs aus mehreren Kernen, sodass die verschiedenen Tasks auf sie aufgeteilt werden können.
- **SSE/SSE2-Befehle (SSE/SSE2 commands):** Auf den ersten Blick scheinen sich diese Befehle perfekt für die Parallelisierung des Summierungsprozesses und die Beschleunigung der Software ohne den Einsatz einer thread-basierten Programmierung zu eignen. Leider basiert der gesamte SSE-Befehlssatz jedoch auf Daten desselben Typs. Da die aufgezeichneten Daten 8 Bit lang sind und der Zwischenspeicher für die Mittelwertbildung 32 Bit lang ist, bietet sich diese Lösung hier nicht an.

► **White Paper**

Ergebnisse

Sämtliche Messungen werden mithilfe eines Digitizers mit 1-Kanal-Abtastung mit 5 GS/s, einer 8-Bit-Auflösung und einem externen Trigger durchgeführt. In der Tabelle sind auch verschiedene Programmeinstellungen aufgeführt, um die Unterschiede der Ergebnisse zu veranschaulichen. Das beste Ergebnis je Segmentgröße ist in der Tabelle gelb hinterlegt.

Abtastrate	Segmentgröße	Mittelungen	Notifysize	Modus	Threads	Max. Triggerrate	CPU Load
5 GS/s	32 kSamples	1000	1 MByte	Hardware	-	150 kHz	< 5%
5 GS/s	128 kSamples	1000	1 MByte	Hardware	-	38 kHz	< 5%
5 GS/s	256 kSamples	1000	256 kByte	Software	2	10.3 kHz	25%
5 GS/s	256 kSamples	1000	1 MByte	Software	2	12.6 kHz	17%
5 GS/s	256 kSamples	1000	1 MByte	Software	4	12.8 kHz	16%
5 GS/s	256 kSamples	1000	1 MByte	Software	-	6.4 kHz	14%
5 GS/s	512 kSamples	1000	512 kByte	Software	2	5.9 kHz	25%
5 GS/s	512 kSamples	1000	512 kByte	Software	4	6.0 kHz	29%
5 GS/s	512 kSamples	1000	1 MByte	Software	4	6.4 kHz	23%
5 GS/s	512 kSamples	1000	2 MByte	Software	4	6.4 kHz	23%
5 GS/s	512 kSamples	1000	8 MByte	Software	4	6.4 kHz	14%
5 GS/s	512 kSamples	1000	8 MByte	Software	-	3.4 kHz	14%
5 GS/s	1 MSamples	1000	1 MByte	Software	-	1.5 kHz	16%
5 GS/s	1 MSamples	1000	1 MByte	Software	2	2.9 kHz	24%
5 GS/s	1 MSamples	1000	1 MByte	Software	4	2.9 kHz	23%
5 GS/s	1 MSamples	100	1 MByte	Software	4	2.9 kHz	30%
5 GS/s	1 MSamples	10000	1 MByte	Software	4	2.9 kHz	23%
5 GS/s	2 MSamples	1000	2 MByte	Software	-	0.7 kHz	14%
5 GS/s	2 MSamples	1000	2 MByte	Software	4	1.3 kHz	40%
5 GS/s	4 MSamples	1000	4 MByte	Software	-	340 Hz	15%
5 GS/s	4 MSamples	1000	4 MByte	Software	2	410 Hz	24%
5 GS/s	4 MSamples	1000	4 MByte	Software	4	390 Hz	50%
5 GS/s	8 MSamples	1000	8 MByte	Software	-	160 Hz	14%
5 GS/s	8 MSamples	1000	8 MByte	Software	2	190 Hz	35%

Zusammenfassung

Die vorstehenden Ergebnisse zeigen, dass die blockweise Mittelwertbildung in einer Software zur Verbesserung der Segmentgröße insgesamt verwendet werden kann, solange die Wiederholungsrate nicht zu hoch wird. Dank der hohen Datenübertragungsraten auf dem PCIe-Bus lassen sich Mittelwerte für wesentlich längere Aufzeichnungen bilden. Dadurch kann eine der Hauptbeschränkungen FPGA-basierter Prozesse für die Mittelwertbildung überwunden werden. In Situationen, in denen extrem hohe Wiederholungsraten verwaltet werden müssen, ist die blockweise Mittelwertbildung nach wie vor die beste Option.

Das besprochene Testprogramm steht frei für eigene Tests oder als Grundlage für die

► White Paper

Implementierung in anderen Softwareprogrammen zur Verfügung.

Die beste Leistung lässt sich mit einem Wert von 1 MB für den Parameter „Benachrichtigungsgröße“ (Notifysize) erzielen. Die Anzahl der durchgeführten Mittelwertbildungen hat keine sichtbare Auswirkung auf die Testergebnisse. Die für das Kopieren des Ergebnissegments und das Löschen des Ergebniswischenspeichers erforderliche Zeit ist im Vergleich zur Summierung der Abtastungen irrelevant.

Da der gesamte Prozess der Datenaufbereitung und Summierung bei Aufzeichnung mehrerer Kanäle nicht abweicht, kann das Ergebnis für andere Kombinationen von Kanälen einfach nachberechnet werden. Sämtliche folgenden Einstellungen führen zur selben maximalen Triggerrate:

- 1 Kanal 5 GS/s @ Segmentgröße
- 2 Kanäle 2.5 GS/s @ Segmentgröße/2
- 4 Kanäle 1.25 GS/s @ Segmentgröße/4

Durch Verringerung der Abtastrate für einen Kanal auf 2,5 GS/s kann ein Kanal mit der maximalen theoretischen Geschwindigkeit der Software zur Mittelwertbildung betrieben werden. Bei einer Segmentgröße von 1 MSample und einer Totzeit von 160 Abtastungen liegt die maximale Triggerrate bei:

$$(2.5 \text{ GS/s}) / (1 \text{ MSample} + 160) = 2.38 \text{ kHz}$$

Das ist deutlich weniger als das gemessene Maximum von 2,9 kHz bei 5 GS/s.