**H ö h e r e T e c h n i s c h e B u n d e s l e h r a n s t a l t**

**S a l z b u r g**

**Abteilung für Elektronik**

**Übungen im**

**Laboratorium für Elektronik**

**Protokoll**

**für die Übung SreS 02**

**Gegenstand der Übung**

|  |
| --- |
| **Logikanalysator** |
|  |

|  |  |
| --- | --- |
| **Name:** | **Clemens Hütter** |
| **Jahrgang:** | **4AHEL** |
| **Gruppe Nr.:** | **B01** |
| **Übung am:** | **09.10.2019** |

|  |  |
| --- | --- |
| **Anwesende:** | Clemens Hütter, Christian Kreidenhuber |

***Inhaltsverzeichnis***

[1. Einleitung 3](#_Toc22074748)

[2. Inventarliste 4](#_Toc22074749)

[3. Übungsdurchführung 5](#_Toc22074750)

[3.1. Training an der Messumgebung 5](#_Toc22074751)

[3.2. Triggerung von Signalen 7](#_Toc22074752)

[3.3. Auflösung und Speichertiefe 11](#_Toc22074753)

[4. Zusammenfassung 14](#_Toc22074754)

# Einleitung

In dieser Übung haben wir Logikanalysatoren und die dazugehörige Software kennengelernt. Um eine Messgrundlage bereitzustellen, haben wir mithilfe eines Crumb Moduls und der darin vorhandenen Timer-Funktion einen 8-Bit-Zähler realisiert. Dieser Zählt mit 1kHz am LSB hoch, bis er insgesamt den Wert 127 erreicht.

Der Timer funktioniert folgendermaßen:

Es wird stätig ein Zählregister hochgezählt. Dieses Register zählt mit der Taktfrequenz des Mikrocontrollers dividiert durch einen Prescaler. Der Prescaler kann beispielsweise Werte von 1, 8, 64, 256, 1024 annehmen.

Im CTC Modus zählt das Zählregister nur bis zu einem bestimmten Wert und setzt sich danach zurück. Dieser Wert wird im Compareregister gespeichert.

Beim Erreichen des Compare-Wertes wird außerdem ein Interrupt ausgelöst.

Um nun noch die richtige Frequenz am LSB zu erreichen, müssen entsprechende Werte für Prescaler und Compareregister gefunden werden. Diese Werte sollten eine möglichst hohe Auflösung bieten. Das bedeutet, dass der Wert des Compareregisters so hoch und der Prescaler so niedrig wie möglich gewählt werden sollen.

In unserem Fall haben wir bereits eine vollständige Bibliothek für Timer erstellt, welche automatisch die Werte für Compareregister und Prescaler mit maximaler Auflösung definiert. Dafür muss in der TimerInit-Funktion der Modus des Timers, also CTC, die Taktfrequenz des Mikrocontrollers und der gewünschte Interrupt-Intervall weitergegeben werden.

Ist der Timer richtig eingestellt, muss nur noch bei jedem Interrupt das Portregister des gewünschten Ausgangs für den 8-Bit-Zähler inkrementiert werden.Ist danach der Wert 127 überschritten, wird stattdessen auf 0 zurückgesetzt.

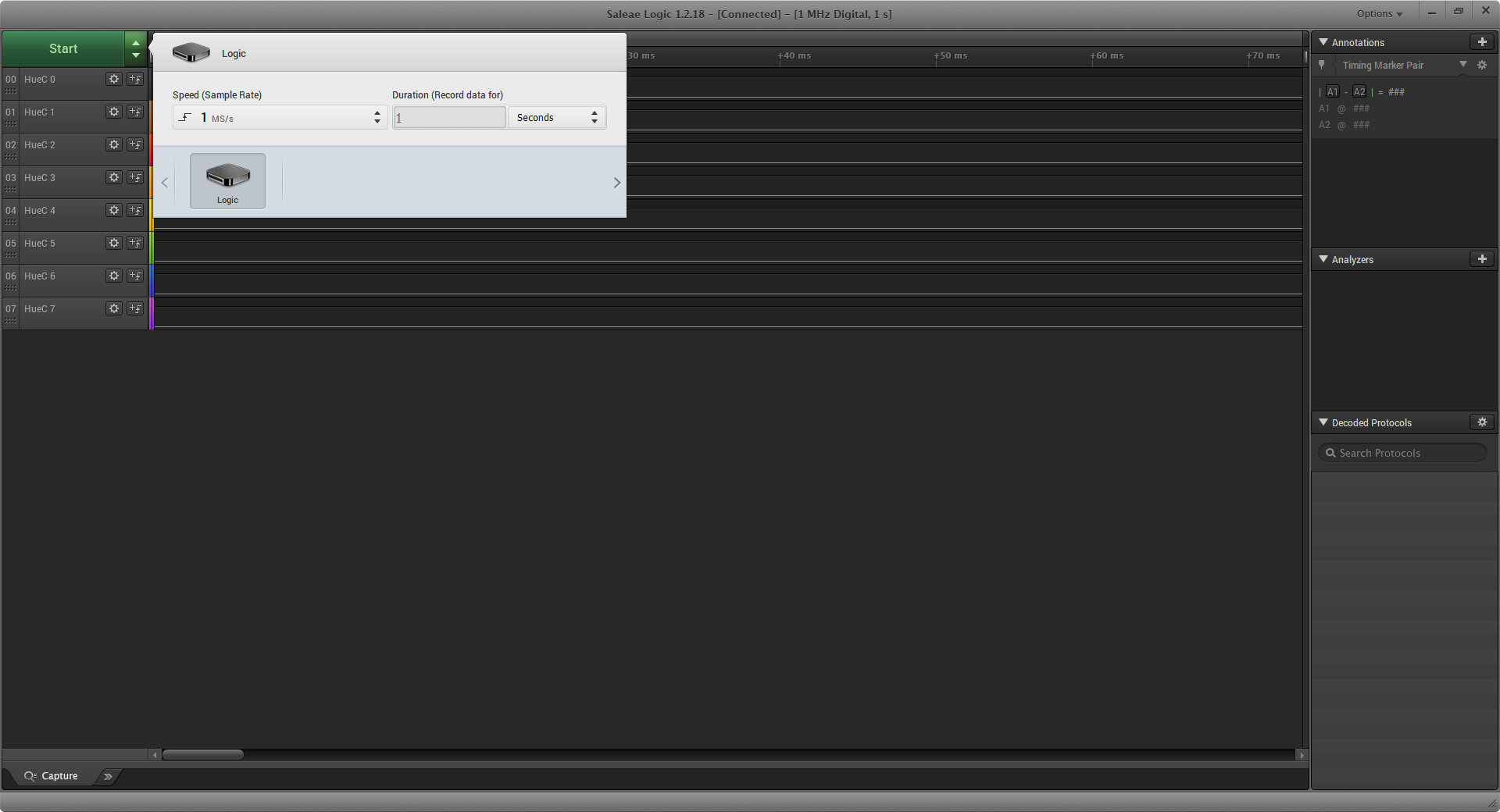
# Inventarliste

|  |  |  |
| --- | --- | --- |
| **Stück** | **Gerätebezeichnung** | **Inventarnummer** |
| 1 | Logikanalysator | Nicht vorhanden |
| 1 | Crumb644 | Nicht vorhanden |

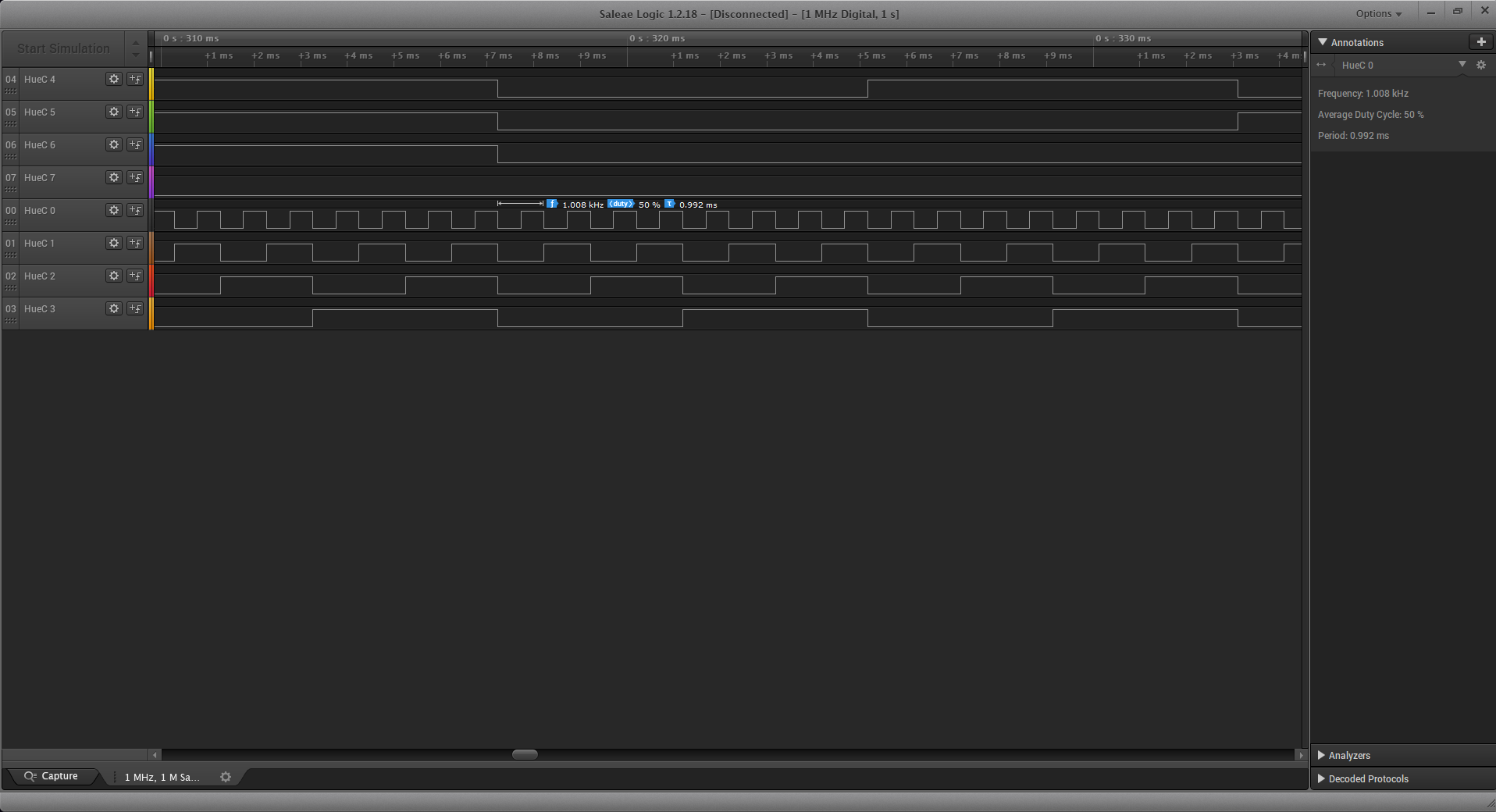
# Übungsdurchführung

## Training an der Messumgebung

Im ersten Teil der Messübung soll man sich zuallererst mit dem Messwerkzeug und der Software vertraut machen. Dafür muss man zuvor alle Pins des Zähler-Ports am Crumb mit den entsprechenden Pins am Logikanalysator verbinden. Dabei ist zu beachten, dass bei der chinesischen Kopie des Saleae Analysators die Pins nicht mit Pin 0-7, sondern mit Pin 1-8 beschriftet sind. Außerdem muss der Analysator noch mit Masse verbunden werden. Danach kann schon eine Verbindung via USB hergestellt werden.

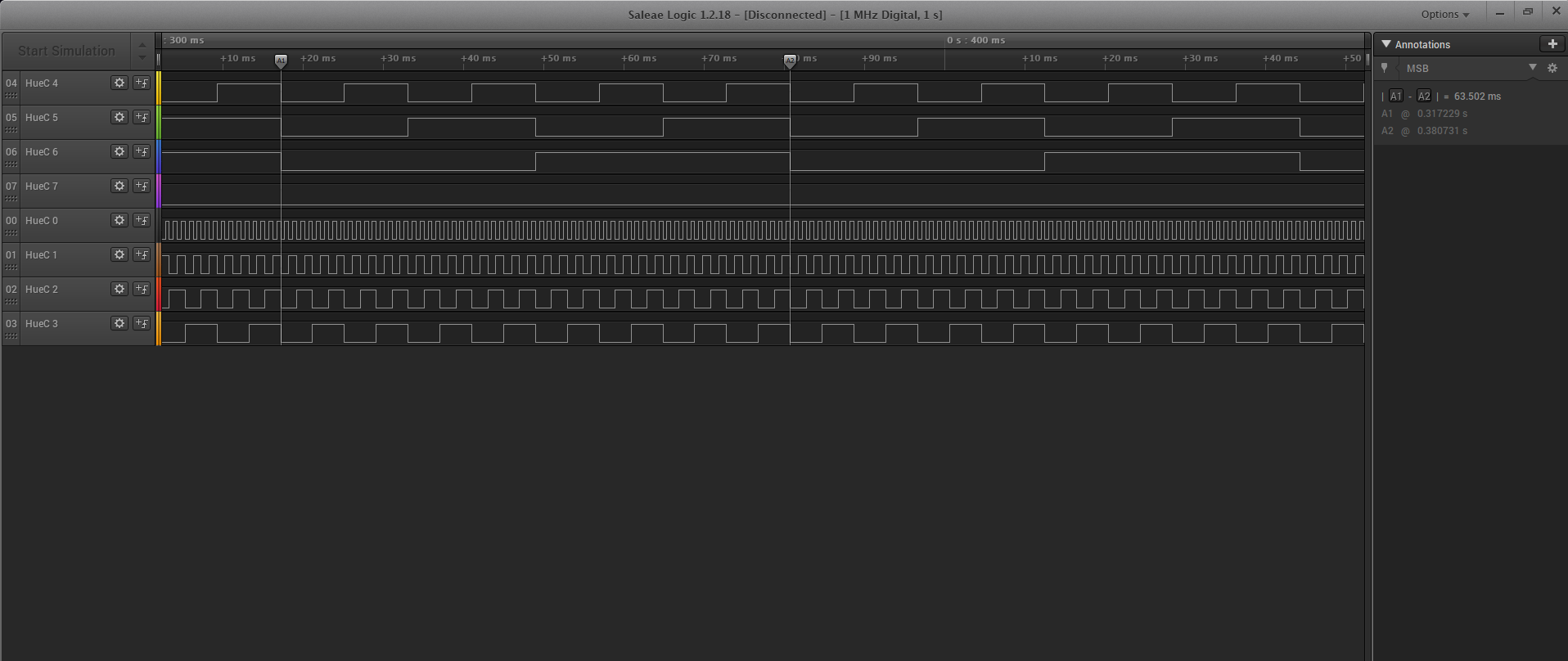


Der nächste Schritt wird in der Software getan. Hier muss die Sampling-Rate festgelegt werden. Das bedeutet, wie oft in der Sekunde abgetastet wird.

Rechts zu sehen ist die Periodendauer und Frequenz des LSB. Die gemessene Frequenz stimmt mit sehr geringer Abweichung mit der eingestellten überein.

In der Software ist es außerdem möglich, alle Eingänge umzubenennen.

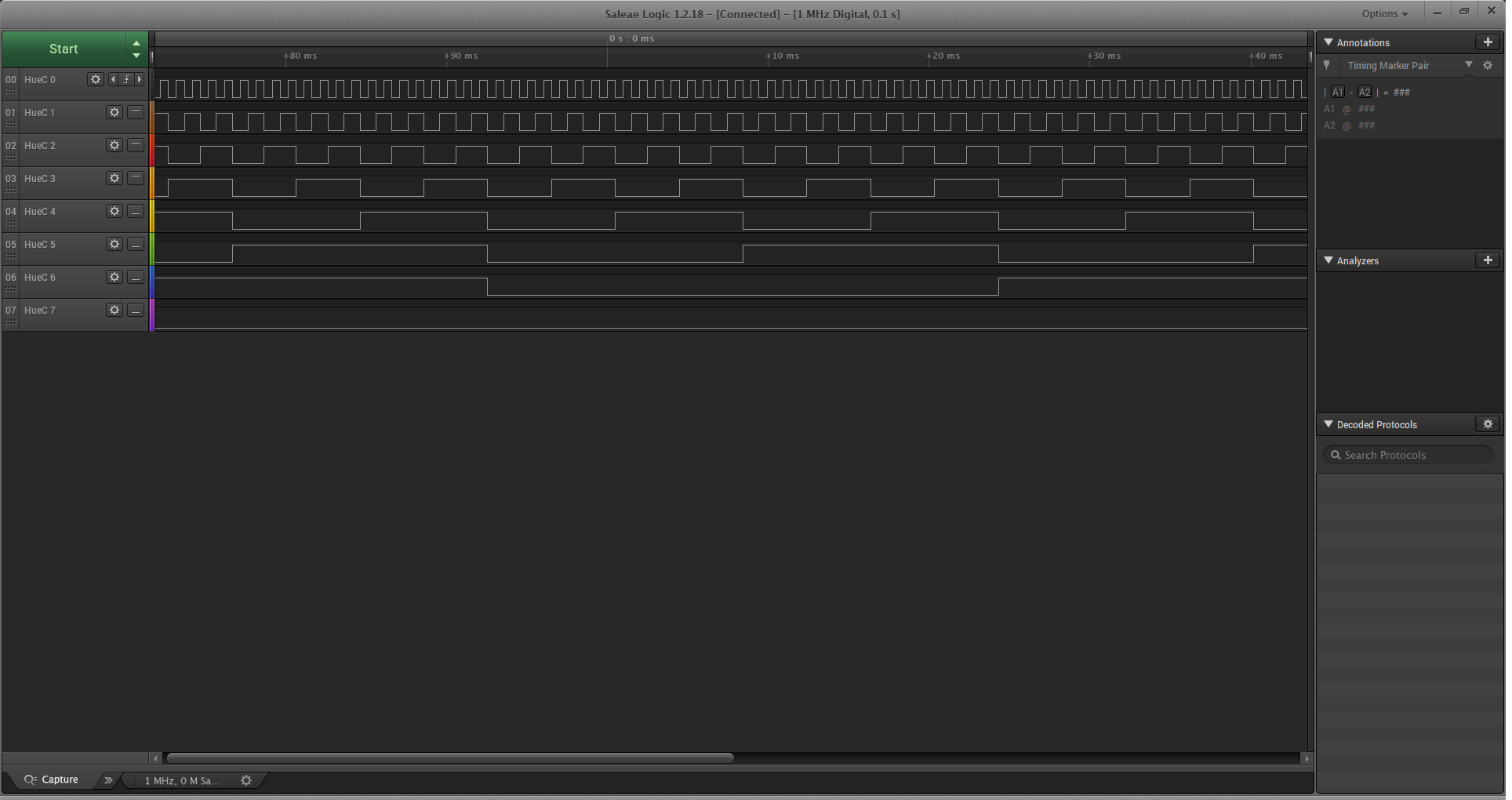
Durch einfaches Drag and Drop kann man die Kanäle auch beliebig neu anordnen. In diesem Beispiel wurden das obere und untere Nibble vertauscht.



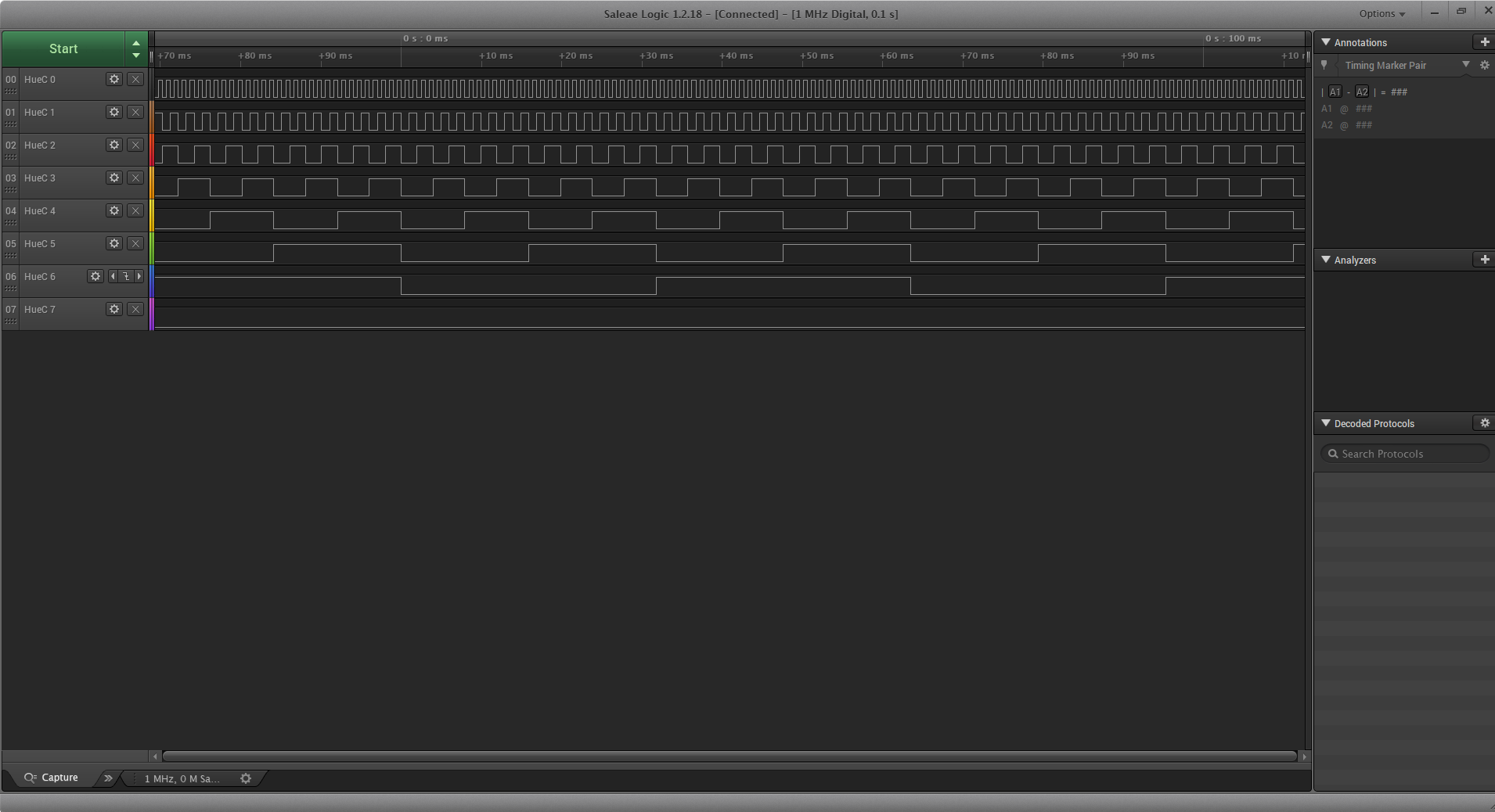
Mit Markern lassen sich in einem Zeitbereich bestimmte Messungen anstellen. Hier wurde eine Periode des MSB gemessen. Mithilfe von snap-to-edge erwischt man auch genau die Flanke. Diese Funktion ist in der neuesten Software Version bereits voreingestellt.

## Triggerung von Signalen

Die Software lässt außerdem zu, dass die Messung erst bei einem bestimmten Ereignis anfängt. Dies nennt man Trigger. Ein Trigger kann bei steigender Flanke, fallender Flanke, High-Pegel, Low-Pegel und Any-Edge ausgelöst werden. Für jeden Trigger lassen sich in der neusten Sofware Version auch zusätzliche Bedingungen festlegen. Damit kann man beispielsweise Festlegen, dass erst getriggert werden soll, wenn alle Kanäle auf High sind.

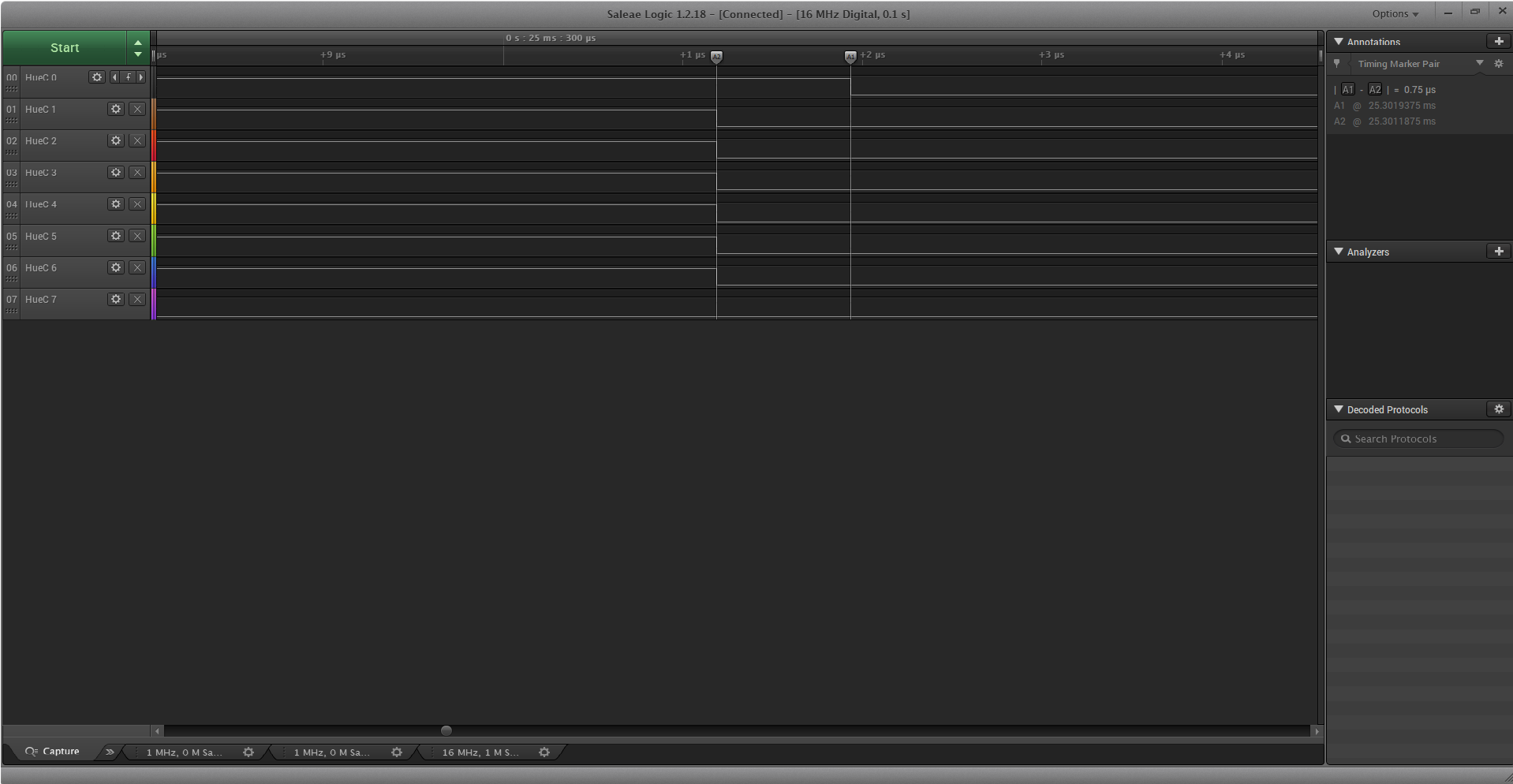


Beispielsweise wurde hier mit steigender Flanke beim LBS getriggert. Deswegen ist direkt bei 0s 0ms eine steigende Flanke zu sehen.

In diesem Fall wurde bei der fallenden Flanke des MSB getriggert. Das sieht man dadurch, dass auf der Zeitskala bei 0s und 0ms eine fallende Flanke des MSB ist. Dadurch haben wir auch erreicht, dass die Messung den Wechsel von 0x7F auf 0x00 darstellt und somit bei Rücksetzen des Zählers beginnt.

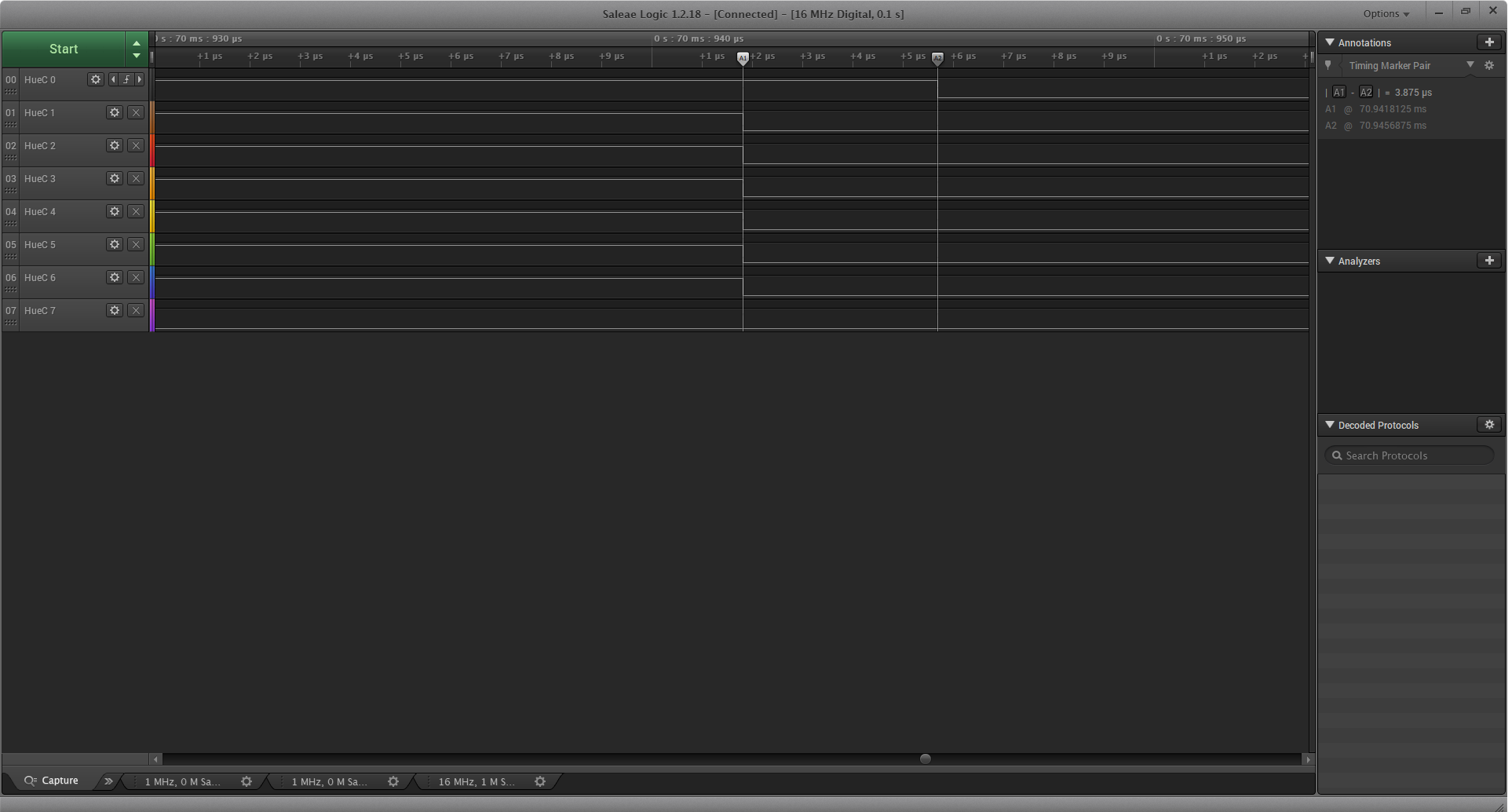
Um die Triggerung für unsere Gruppennummer \* 16 – 1 = 15 einzustellen, muss bei einem der Bits des unteren Nibbles als Trigger die steigende Flanke festgelegt werden. Für die anderen 3 Bits sollte die Zusatzbedingung für den High Pegel eingestellt werden. Dadurch wird genau beim Wert 15 getriggert.

Zunächst haben wir an das LSB des Zählers eine Kapazität angeschlossen. Als erstes haben wir einen Kondensator mit 22nF benutzt.



Dadurch verändert sich die Laufzeit in dem Sinne, dass dieses Bit länger für einen Pegelwechsel von High zu Low benötigt als die anderen. Dadurch sieht man dass Leitungskapazitäten äußerst unerwünschte Störgrößen sind, welche zu großen Problemen führen können.

Danach haben wir noch einen 100nF Kondensator angeschlossen.

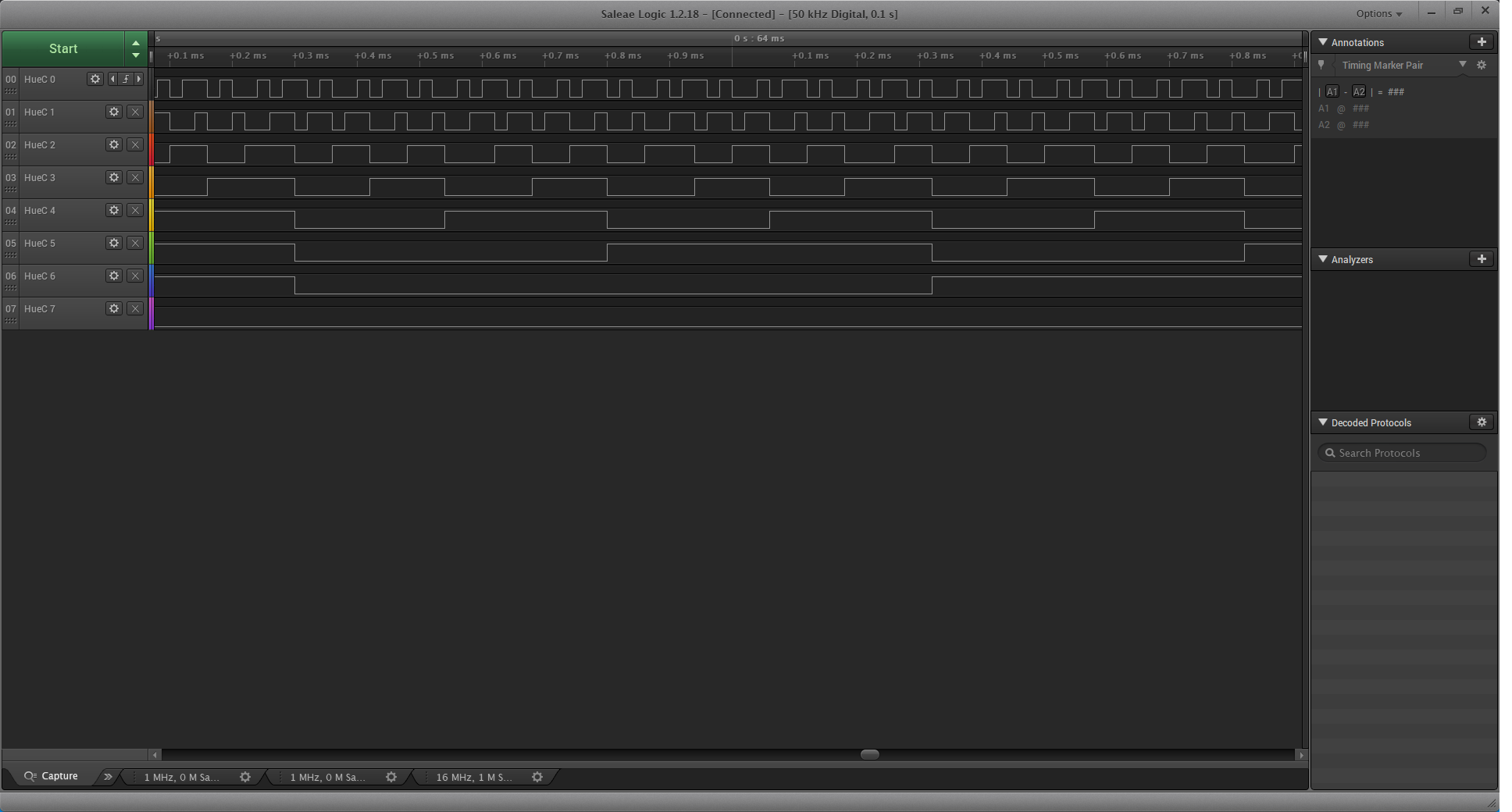


Durch eine Vergrößerung der parasitären Kapazität werden auch die Lautzeiteffekte größer.

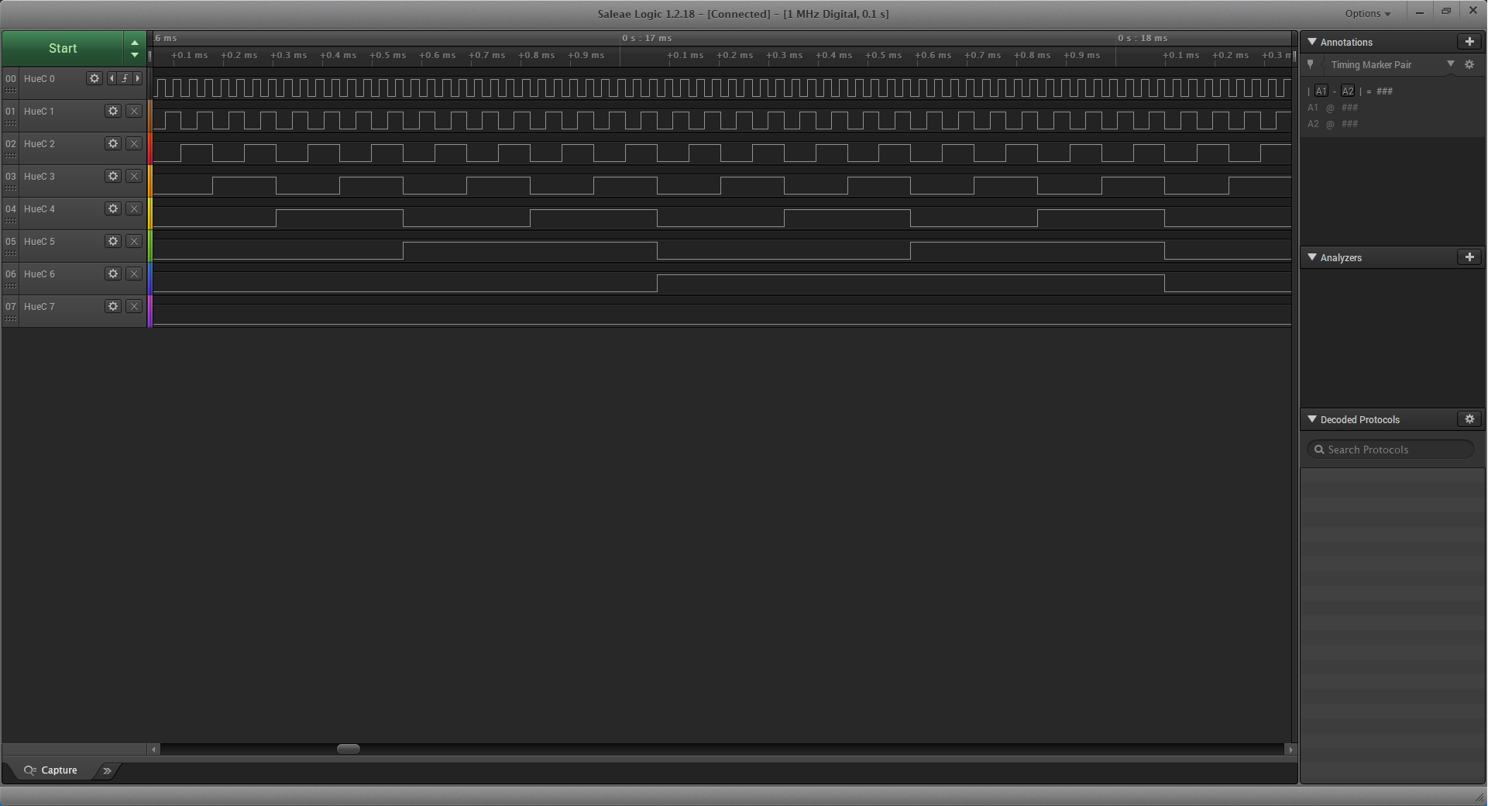
## Auflösung und Speichertiefe

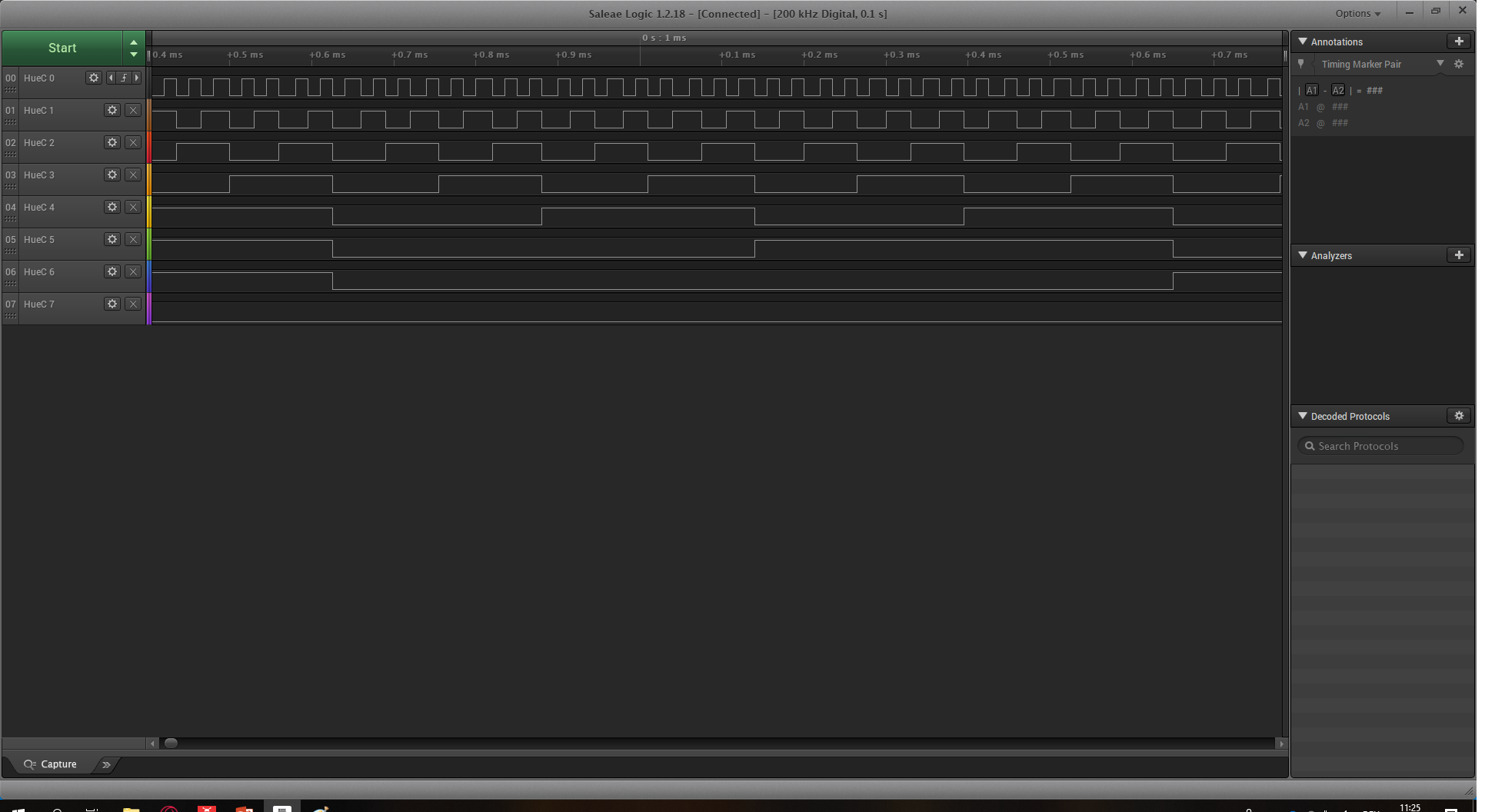
Ist die Sampling Rate, also die Abtastrate zu hoch ist der Speicher äußerst schnell voll. Ist sie zu niedrig, wird das Signal fehlerhaft dargestellt.

Für eine ordentliche Messung ist es also wichtig, den richtigen Mittelweg zwischen Messqualität und Messdauer zu finden.



In diesem Bild wurde mit 50kHz Abtastrate gemessen, was offensichtlich zu wenig ist, da die niederwertigen Bits nicht gut genug gemessen werden können. Zu erkennen ist ein nicht periodisches Signal mit willkürlichem Duty-Cycle. Höhere Bits, die mit niedrigerer Frequenz schwingen, werden allerdings originalgetreu dargestellt.

Durch Erhöhung der Abtastrate auf 1MHz verschwinden die zuvor bemängelten Fehler, sie nun entsprechend der Zählfrequenz hoch genug ist. Dadurch lässt sich auch bei den niederwertigen Bits wieder ein Duty-Cycle von 50% feststellen.

Durch sukzessives Reduzieren der Abtastrate kamen wir auf das Ergebnis, dass ab einer Sample Rate von etwa 200kHz die Messung des LSB, welches mit 1kHz schwingt, beginnt, ungenau zu werden.

# Zusammenfassung

Wie unterscheidet sich ein Logikanalysator von einem DSO?

Zuallererst bietet ein Logikanalysator im Gegensatz zu einem DSO 8 Kanäle als Eingang, während ein DSO meistens zwischen 2 und 4 besitzt. Außerdem stellt ein DSO den genauen Spannungswert des Eingangskanals dar, während ein Logikanalysator lediglich zwischen High und Low Pegeln unterscheidet. Zuletzt ist ein kleiner Unterschied, dass DSOs vorrangig als eigene Geräte, sozusagen als Standalones, existieren, während ein Logikanalysator via USB und zusätzlicher Software an einen PC gebunden ist.

Was versteht man unter der „snap to edge“ Funktion des Cursors?

Mit der snap-to-edge Funktion ist es möglich, einen Cursor pixelgenau auf einer Flanke zu platzieren, da der Computer ihn automatisch an eine erkannte Flanke anheftet.

Unterschrift:

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum:** | **Note:** | **Punkte:** | **Unterschrift:** |