

Mätlaboration

Introduktion till Digital Discovery

LiU - ISY - Datorteknik

27 september 2022

1 Inledning

Logikanalysatorn är ett av de absolut mest användbara verktygen vid felsökning av digital elektronik. Syftet med den här laborationen är att du ska kunna använda en logikanalysator effektivt för verifiering och felsökning.

Det finns inga förberedelseuppgifter, men läs gärna igenom denna laborationsmanual i förväg. Det innehåller principer för mätning med en logikanalysator. Avsnitt 2 beskriver laborationsuppställningens inkoppling av mätobjekt och logikanalysator och avslutas i avsnitt 3 med ett antal övningar och frågor. När du tycker att du är klar med dessa, eller på annat sätt känner dig bekväm med mätmiljön, så kan du redovisa. Tanken med övningen är att alla ska få chansen att experimentera med en logikanalysator med hjälp nära till hands.

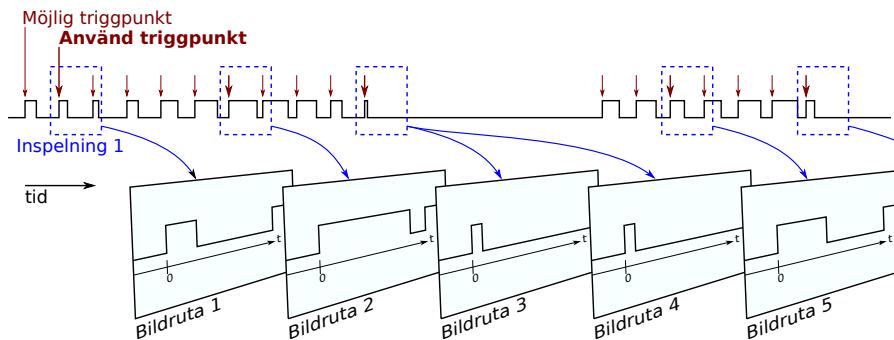
1.1 Triggning

Ett centralt begrepp är *trigga*, vilket innebär att definiera "tidpunkt 0" för en given mätning som ska visas på skärmen. Triggvillkoret kan t.ex. vara att en signal får en stigande flank (övergång från 0 till 1) eller startbit på någon databuss, eller något annat.

Analysatorn lyssnar på sladdarna, spelar in i en cyklisk buffert, och avgör när ett triggvillkor är uppfyllt ("triggpunkt"). Då fortsätter inspelningen en kort stund, och resultatet kopieras till skärmen. Efter det fortsätter analysatorn att lyssna efter en ny triggpunkt. Den inspelade signalen kring triggpunkten kopieras till skärmen så ofta att det ser kontinuerligt ut för oss.

Genom att ändra tidsskalan kan man zooma in på en intressant del av en signal. Eventuellt måste man då lägga triggpunkten utanför den synliga delen av signalen.

Figur 1 visar ett exempel på en mätning av en pulsbreddsmodulerad signal i 270 Hz med triggas på stigande flank. Skärmen har en uppdateringsfrekvens på 60 bildrutor/s och visar bildrutorna illustrerade i figuren. Triggpunkterna i figuren kommer oftare än bilduppdateringsfrekvensen vilket gör att det bara är var 4:e eller 5:e triggpunkt som används. Eftersom signalen ibland stannar av så har ingen ny inspelning gjorts när det är dags för bildruta 4 att visas och då kommer inspelningen till bildruta 3 att visas igen. I figuren har zoomläget anpassats så att drygt en hel period av signalen syns på skärmen.



Figur 1: Princip för mätning med logikanalysator.

1.2 Sampelfrekvens

Logikanalysatorn *Digital Discovery* har en sampelfrekvens på 100 MHz (går att ställa om). Denna kan användas för att identifiera triggpunkter.

Däremot har inspelningsbufferten en upplösning som beror på hur inzoomad skärmen är, och förmodligen betydligt lägre än sampelfrekvensen.

Förvirring 1: Väldigt korta störningar kan orsaka en triggpunkt, som dock inte syns på skärmen ifall de glesare samplerna i inspelningen missat den.

Förvirring 2: Man kan stoppa mätningen, för att kunna undersöka aktuell inspelning. Om man då zoomar in på någon detalj, så kommer den begränsade upplösningen i bufferten att framstå som “sneda övergångar” mellan ettor och nollor. Dessa sneda övergångar har inget med analog spänning att göra.

2 Uppkoppling

En kortfattad beskrivning av laborationsuppställningens grundläggande uppkoppling av mätobjekt och logikanalysator följer.

2.1 Mätobjektet

Vi ska göra mätningar på den ATmega16 som sitter på vår vanliga labbmiljö. För det behöver du flasha in aktuellt program på den:

1. Ladda ner filen “Mat-labb.hex” från kurshemsidan.
2. Starta Atmel Studio.
3. I Atmel Studio, välj *Tools* → *Device Programming*.
4. Välj Tool till JTAGICE3, Device till ATmega16A och Interface till JTAG. *Apply*. Du glömde väl inte att slå på strömmen på kortet?
5. Fliken *Memories*.
6. I rutan Flash, välj den nedladdade .hex-filen. *Program*.
7. Så fort nedladdningen är klar, så ska lysdioderna A7 och A5 blinka två gånger. Missade du det, så prova att trycka på Reset-knappen.
8. Stäng Atmel Studio.

Det blir din uppgift att ta reda på vad som händer på port D. Knaptryckningar på INT0, INT1 och INT2 påverkar beteendet.

2.2 Grundläggande inkoppling

Logikanalysatorns probar (kontakter) är anpassade för en annan sorts pinnar än vad labbplattan har. Därför har du fått en adapter. Fråga en laborationsassistent om du är osäker.

Koppla in någon av logikanalysatorns svarta sladdar till Gnd på labbplattan. Koppla in logikanalysatorns kanaler 0–7 till pinnarna PD0–PD7 (notera ordningen). Koppla även in kanal 8 till pinne PB2 — detta mäter INT2.

Var försiktig med hårdvaran. Spända sladdar kan gå sönder, med glapp och felaktiga mätningar som resultat.

2.3 Analysatorn

Vi använder logikanalysatorn *Digital Discovery*, som består av en liten “logikpuck” som är kopplad till datorn, samt programmet WaveForms (se <https://reference.digilentinc.com/reference/software/waveforms/waveforms-3/start>).

Starta WaveForms (W) från startmenyn på datorn.

Du måste nu skapa ett nytt “workspace”: I tabben *Welcome* väljer du *Logic*, och får upp en ny tabb som heter “Logic 1”.

Klicka på det gröna korset för att lägga till kanal. Lägg till “Signal”, och välj DIN 0–7. Nu ska DIN 0 vara kopplad till PD0 på kortet etc. Klicka på “Run”. Du ska se ett kaos bland signalerna på skärmen.

3 Mätuppgifter

Nu är det dags att börja mäta. Du är i ganska hög grad utlämnad till att testa dig fram, med endast några enstaka tips. Var rädd om fysiska sladdar, men var inte rädd att prova dig fram i mjukvaran.

Om du stöter på problem, försök att googla eller lösa det själv (det är en bra träning i problemlösning, samt att man ofta lär sig bättre så) annars finns labbassistenten nära till hands.

3.1 Hitta alla “aktiva” signaler

Du bör direkt på skärmen kunna avgöra vilka signaler som det händer något på. INT0-INT2 på labbplattan ska påverka beteendet på några av pinnarna, försök att se hur.

3.2 Mäta tid

Tryck på “Stop” för att stanna mätningen. Zooma (skrollhjulet) så du ser två hela pulser på PD5. Eventuellt kan du behöva köra ett Single-steg (bredvid Run-knappen).

Kolla på X-axeln och försök att ungefärligt uppskatta periodtid (stigande till stigande flank) och pulstid (stigande till fallande flank) på PD5. Sedan kan du mäta upp dessa exakt – till höger direkt ovanför vågfönstret så finns några ikoner som kan hjälpa dig.

Vad har PD4 för “duty cycle” (procent av periodtiden som signalen är hög)? Vad har PD4 för frekvens?

3.3 Trigga på något

Starta mätningen igen (Run).

Utan triggvillkor så ser signalerna bara kaosartade ut. Genom att klicka på “X” i kolumnen T, så kan du skapa en “simple trigger”. Prova att trigga på stigande flank på PD4 och PD7, en i taget.

Prova sedan att istället trigga på stigande flank på PD5. Kan du nu se vad INT0 och INT1 har för effekt på PD5?

3.4 Zooma och panorera

När du triggat på något, så kan du zooma med skrollhjulet, och panorera genom att dra i vågfönstret med muspekaren. Notera att triggpunkten markeras med en svart pil på x -axeln. Du kan även ställa in zoom-nivå och offset i textfälten uppe till höger. Om du råkat zoomat bort från triggpunkten, så kan du sätta Position=0 uppe i högra hörnet, t.ex. genom att skriva en nolla i rutan.

Du kan prova att trigga på fallande flank på PD2 och PD3, och zooma så du ser triggpunkten och 2-3 ms framåt. Klicka på INT0 och INT1 upprepade gånger. Du bör nu se några fall av så kallade kontaktstudsar.

Kontaktstudsar: En resistor drar sladden upp mot 5V, men den är ganska svag. Knappen kortsluter signalen mot jord, men det är en mekanisk knapp där två metallbitar förs ihop. Då studsar de till lite under någon millisekund, och resultatet blir just det du ser.

Detta orsakar att en tryckning på knapparna kan detekteras flera gånger, vilket kan ställa till det.

3.5 Trigga (forts)

PD0 skickar ut korta pulser, värda att titta närmare på. Trigga på stigande flank på PD0, och zooma in på pulsen. Den verkar variera mellan två bredder, lyckas du utläsa hur breda?

Prova att enbart trigga på den lite längre av pulserna: Bland knapparna ovanför vågfönstret så finns triggknappen “Pulse”. Välj alternativet “More”, och ställ in lämplig gräns, någonstans emellan den korta och den långa i längd.

Har du hittat att ett tryck på INT2 ger ett pulståg på PD1 än? Ändra triggern till att detektera en när PD1 varit '0' i minst 10 μ s. Zooma in på pulståget. Tryck på INT2 några gånger för att se att vågformen uppdateras varje gång.

När du inte tryckt på någon knapp, så ska pulståget tystna, och efter en stund så “tröttnar” WaveForm på att vänta efter ny triggpunkt, och börjar visa “kaoset” igen.

Ändra Trigger mode (drop-down till höger om Run-knappen) från “Auto” till “Normal”, vad händer nu när ingen triggpunkt kommer?

3.6 Avkoda parallellbuss

PD4 till PD7 ser ut som en räknare, med undantag för att PD5 håller på med något annat. För att få en bättre bild av värdet, kan vi tolka bitarna binärt.

Lägg till dessa som en ny kanal: Gröna plus-tecknet - *Bus*. Lägg till aktuella bitar (DIN4-7 om du kopplat rätt), och sätt LSB = 0 (d.v.s. PD4).

Trigga, zooma och stoppa sedan när du ser en hel period av PD7. Vad får du för sifferordning under perioden? Hur påverkar PD5 värdet? Prova gärna att single-steppa några gånger för att få in PD5-pulser här och där.

3.7 Servosignal

Triggar åter på stigande flank på PD5, och zoomar så du ser två pulser. Med INT0 och INT1 så ändras pulstiden. Vilken är den kortaste resp. längsta pulstiden du kan få för denna signal?

Signalen är en s.k. servosignal, vilken används för att styra många servon (t.ex. styrservot på bilarna, eller gripklon på Sumo).

3.8 Avkoda UART

Det där pulståget på PD1, som kommer vid INT2, det är en seriell databuss, UART, som skickar ut en text med ca 62.5 kbps. Varje byte består i det här fallet av 10 bitar: En startbit (alltid 0), åtta databitar, och en stoppbit (alltid 1). Ingen paritet.

Lägg till denna som en ny kanal: Gröna plus-tecknet - *UART*. Välj att visa som ASCII.

Trigga på fallande flank på PD1, och tryck sedan på INT2. Panorera/zooma tills du ser den avkodade texten. Man får läsa en bokstav i taget (inom hakparenteserna), så det går lite långsamt.

3.9 Record: Spela in längre stycken

Varje tryck på knappen ger ett slumpmässigt meddelande. Vissa meddelanden är så långa att du nog inte kan läsa hela.

I menyn "Mode", ändra från "Repeated" till "Record". Klicka på Config, ungefär där Run-knappen fanns förut. Det kan vara lämpligt att köra 100k sampler i 5 MHz (då får man även med de korta pulserna på PD0).

Fältet "Trigger" väljer hur stor del av inspelningen som sparas innan triggerpunkten. Om du t.ex. väljer 10%, så kommer triggerpunkten att hamna nästan i början av inspelningen.

När du är klar och klickat på "Record"-knappen, så kommer den att visa löpande data (utan trigg), tills den får en triggpunkt, och då övergår den till att visa inspelningen kring den.

Nu bör du kunna se även de långa meddelandena.

4 Epilog

Nu behärskar du de grundläggande funktionerna med vår logikanalysator. I verkligheten är ju dock spänningar analoga. Exempel: En UART-sändare som stängs av, kan sätta sladden till nivå 5 V (stoppbiten), och sedan släppa den. Sladden kan då långsamt ladda ur sig, och efter ett tag blir spänningen så låg att den tolkas som en nolla. När vi mäter upp det, så ser det ut som om signalen *sätts* till noll efter en viss tid.

4.1 Perspektiv på tid

Vi människor har svårt att uppfatta någonting som är kortare än ett 10-tal millisekunder, medan det är att betrakta som oceaner av tid i elektronksammanhang.

Dock har även elektronik stort omfång av tidsbegrepp. I projektet körs de flesta ATmegor/AVR:er i 16 MHz, så utgå från det. Några illustrationer (alla siffror är ungefärliga):

- **UART i 9600 baud:** Ca 1000 bytes per sekund. Det är lika många som sekunder på en kvart.
- **ljudreflektion 1 meter:** Ca 6 ms. Det motsvarar en tredjedels skärmuppdatering. Går du i 7 km/h, så hinner du 12 mm på den tiden.
- **UART i 1 Mbaud:** Lika många bitar per sekund, som sekunder på tolv dygn.
- **AVR-klockcykler för varje byte i UART 9600 baud:** 16000. Trots att 9600 baud ger betydligt högre överföring än vi människor hinner läsa, så är det väldigt långsamt. AVR:en hinner ta en kafferast mellan varje byte. Eller utföra ett fåtal flyttalsadditioner.
- **Samplhastigheten 100MHz:** 6 sampel för varje klockcykel i AVR:en. Det är ganska lagom.
- **ATmega1284p på flygplan:** Om flyget åker i 720 km/h, så hinner ATmegan utföra 80 klockcykler per färdad mm. Då ska den väl hinna med att hantera en robot som kör i knappt 4 km/h?
- **Intel i7, 4 GHz:** Lika många klockcykler per sekund, som sekunder på 126 år. Och för att skynda på lite, så utförs åtskilliga instruktioner varje klockcykel i varje kärna.