

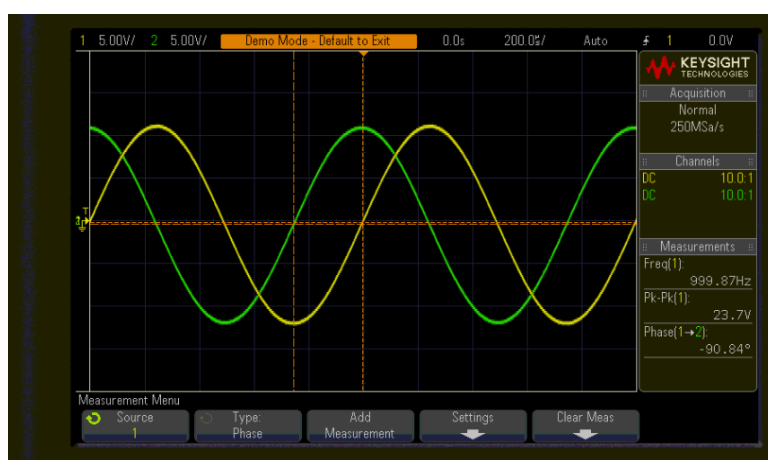
SME Protokol – pokročilé použití osciloskopu

Vojtěch Michal

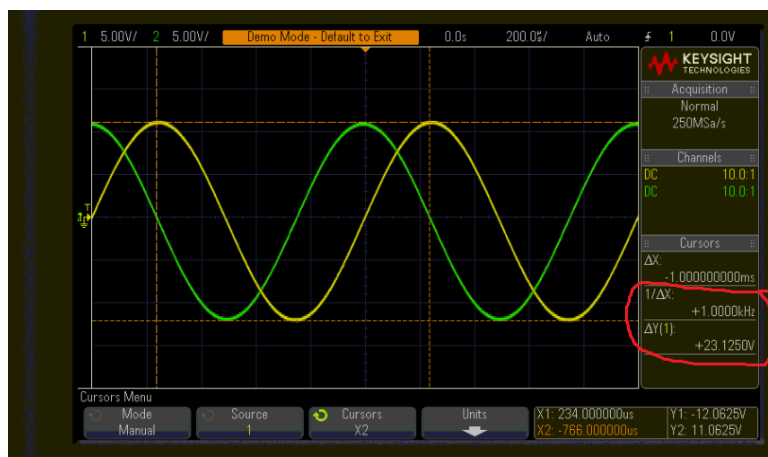
2. května 2021

Úkol 1 Harmonické signály

Úkol 1.1 Měření charakteristik



Obrázek 1: Charakteristiky měřené automaticky



Obrázek 2: Charakteristiky měřené kurzorem

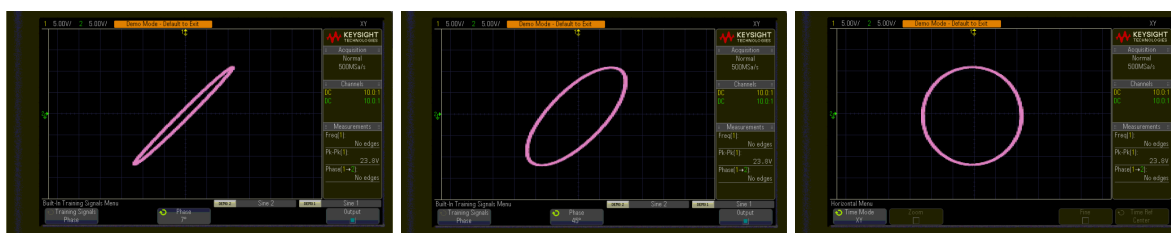
Automatické měření frekvence je závislé na znalosti celé periody signálu, protože je nezbytné trochu zvětšit časovou základnu, zatímco člověk může "tipnout" hodnotu frekvence, ačkoli vidí jen třeba půlperiody. Ve všem ostatním ovšem automatika vede. Protože jsou k dispozici pouze dva kurzory na každou osu (a víc by jich ani nebylo ovladatelných pro nepřehlednost), nelze jednoduše naráz kurzorem měřit frekvenci signálu i jeho fázový posun. Kdyby nebyly signály ideální, ale například by trochu fluktovalo V_{pp} , je automatické měření k nezaplacení – má aktualizovanou hodnotu vypočítanou dříve, než člověk vůbec zpozoruje, že kurzory nejsou na správném místě.

Úkol 1.2 XY zobrazení

Na rozdíl od normálního režimu, kde je na vodorovné ose čas a na svislé ose velikost veličiny, v XY zobrazení se na obě osy vynášejí normovaná hodnota veličin ze dvou kanálů a čas se přímo nevynášejí nikde. V čase t se do zobrazení přidá jeden bod P se souřadnicemi

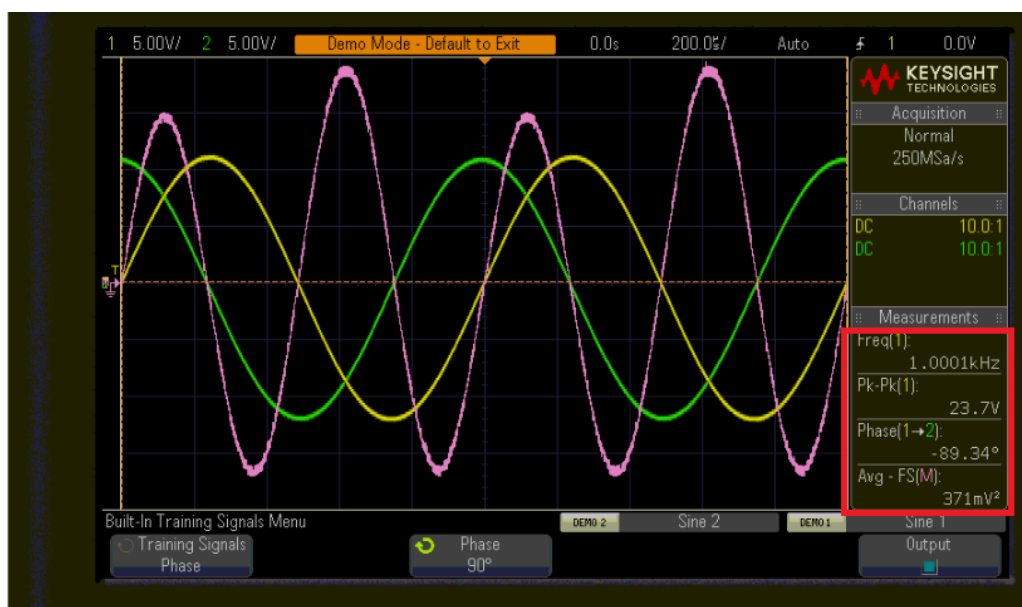
$$P(t) = [x_{center} + u_1(t)y_{center} + u_2(t)] [x_{center} + K_1\sin(\omega_1 t + \phi_1)y_{center} + K_2\sin(\omega_2 t + \phi_2)],$$

kde K_i jsou normovací konstanty pro oba signály, což připomíná parametrizaci elipsy v rovině. Pro $\omega_1 = \omega_2$ (případ na obrázcích) je vykreslený obrazec stacionární a je funkcí pouze $\delta\phi = \phi_2 - \phi_1$. Pro $\Delta\phi = 0$ degeneruje křivka do úsečky na ose I. a III. kvadrantu, pro maximální $\Delta\phi = \frac{\pi}{2}$ je křivka kružnice. Ostatní hodnoty vedou na různě deformovanou elipsu. Užitečné pro sledování fázového rozdílu dvou signálů na stejné frekvenci. Pakliže se frekvence liší, tvar obrazce se v čase mění, aby postihl změny okamžitého fázového posunu.



Obrázek 3: XY zobrazení fázově posunutých signálů. Zleva doprava fázový rozdíl 7°, 45° a 90°

Úkol 1.3 Matematika



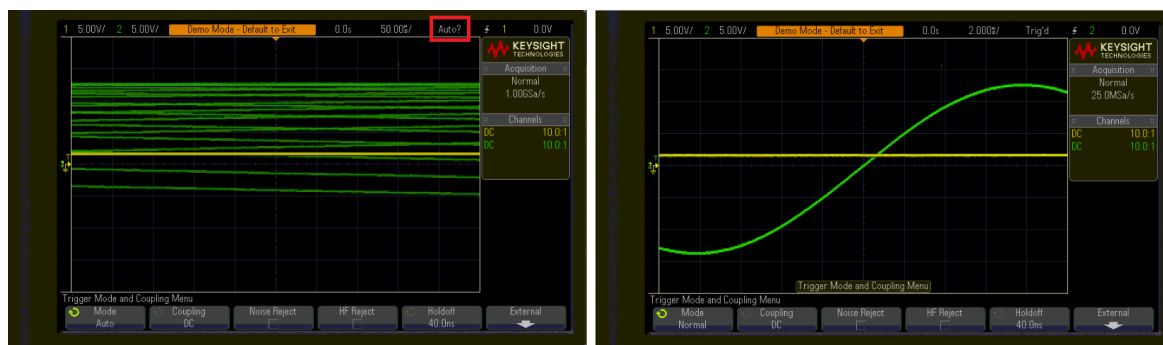
Obrázek 4: Součin dvou posunutých harmonických signálů

Pakliže by harmonické signály představovaly proud a napětí, jejich součin bude odpovídat okamžitému výkonu, což bude harmonický signál na dvojnásobné frekvenci. Integrací přes časový úsek (ideálně periodu) a rozkladem na složky vzniká činný a jalový výkon. Činný výkon bude nulový na prvcích bez ohmického odporu, tedy induktoru a kapacitoru. Konkrétně v případě 4 by mohl být kanál 1 (žlutý) napětím a kanál 2 (zelený) proudem. Poté by matematický kanál představoval výkon na kapacitoru, jehož reálná složka je nulová kvůli fázovému rozdílu I a U o $\frac{\pi}{2}$.

Úkol 2 Trigger

Úkol 2.1 Auto vs normal

Když je na vstupu osciloskopu signál a je splněna trigger condition (například signál projde přednastavenou úrovní ve směru vzestupném/spádném, na který je osciloskop citlivý), jsou oba módy identické. Pokud ale trigger condition splněna není, poté normal mode jen sampleuje a čeká (symbol **Trig'd?** v horním pravém rohu), zatímco autotrigger čas od času pustí vzorkování a vykreslí průběh bez ohledu na trigger condition (viz 5). Protože vzorkování není synchronizováno se signálem, jsou průběhy zachycovány v náhodných okamžicích a na displeji vzniká barevná mlha. Alespoň je ale patrné, že je připojen vůbec nějaký signál.



(a) Auto trigger (bez triggeru)

(b) Normal mode (trigger condition splněna)

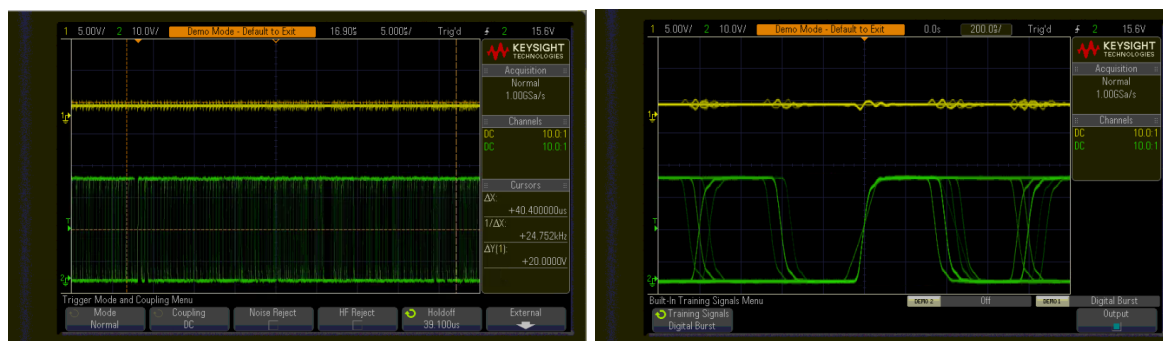
Obrázek 5: Srovnání módů triggeru

Režim auto je proto vhodný pro získání všeobecného přehledu o signálu: zda se vejde na vertikální stupnici, zda se jedná o periodický děj, či jen děj přechodový, atd.

Úkol 2.2 Trigger Holdoff

Nastavení minimální prodlevy dvou po sobě jdoucích triggerů. Po detekování trigger condition začne osciloskop odpočítávat čas do reaktivace triggeru. Nejsnazší je příklad komunikační sběrnice, po které se digitální data přenáší v *rámčích*. Průběh signálu na sběrnici lze interpretovat jako obdélník (sic s proměnlivou periodou). Kdyby byl trigger citlivý na každou hranu, detekuje osciloskop všechny změny logické hodnoty a vykreslené průběhy se budou křížit, neboť by nastávalo více triggerů na různých místech týchž dat. Ilustrační příklad je na obrázku 6b.

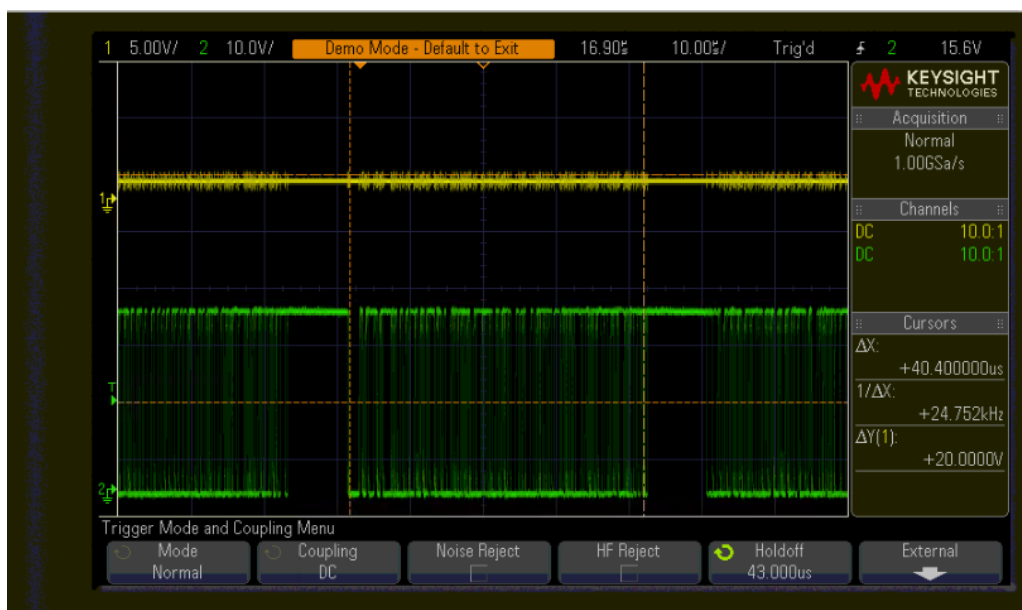
Vložení nenulového pozdržení triggeru zajistí, že osciloskop počká stanovený čas a teprve následně bude hledat trigger condition. Trigger holdoff je potřeba nastavit precizně – nesprávný delay způsobí, že trigger bude rearmnut ještě v témže nebo až uprostřed následujícího rámce. Obě varianty vedou na stejné problémy s křížícími se průběhy, příklad je na 6a.



(a) Nesprávný holdoff, který nestačí na periodu signálu (b) Žádný holdoff (osciloskop vnutí 40ns minimum)

Obrázek 6: Nesprávná nastavení trigger holdoff

Správné nastavení pozdržení triggeru je na délku rámce, aby *trigger rearm* vyšel do "ticha" mezi dvěma framy – například do stop bitu na UARTu. Takové nastavení je vidět na obrázku 7. Protože data posílaná na simulované sběrnici jsou velice proměnlivá, pustil jsem nejprve osciloskop v single módu pro získání jednoho

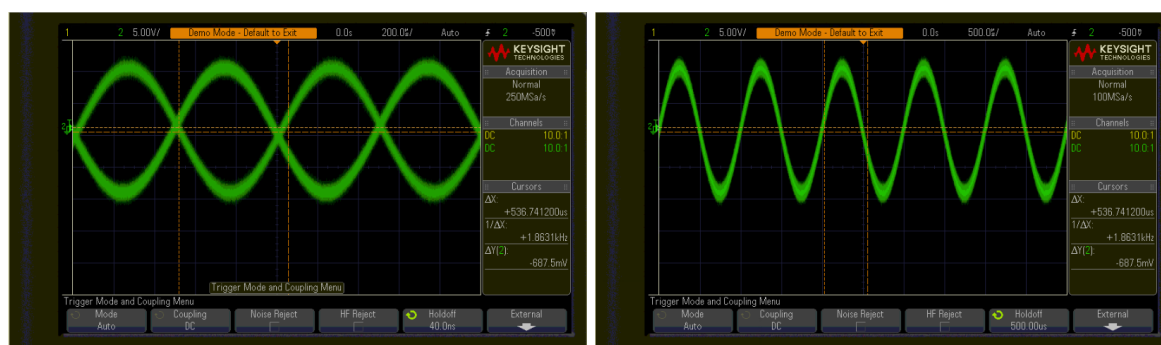


Obrázek 7: Správný holdoff na délku rámce digitálního signálu

nerozmazaného průběhu. Následně jsem změnami časové základny hledal mezery, jež by indikovaly předěl dvou rámců. Kurzory jsem identifikoval pevnou délku rámce 40 μ s, po které následovalo cca 8 μ s prodlevy. Na obrázku je vidět, jak nastavený holdoff 43 μ s garantuje stabilní trigger, při kterém rámce nelitají po celé obrazovce.

Pozn: Pakliže jsou rámce proměnlivé délky, výrazně se situace zkomplikuje. Hodí se myšlenku trochu obrátit a třeba triggerovat na délku pulsu větší než nějaké T . Tato metoda by asi nefungovala pro SPI/UART, ale fungovala by pro CAN. Standard obsahuje *bit stuffing*, jehož důsledkem není možné, aby ve validním framu bylo víc jak pět stejných logických úrovní v řadě. Pakliže by data obsahovala podobný případ, vloží se hardwarově během posílání stuffing bity, aby byl garantován dostatečný počet resynchronizačních hran v signálu. Ve validních datech tedy nikdy nesmí být víc jak pět jedniček v řadě, zatímco *End of frame* jich čítá asi sedm (a případně ještě víc, pakliže by následně měla být sběrnice recesivní). Se znalostí baudratu lze vypočíst minimální šířku okna recesivních úrovní mezi rámci a triggerovat na něj.

Další uplatnění trigger holdoff je pro triggerování zarušených signálů, pro něž by se hodil třeba komparátor s hysterezí. Protože je na "hlavním signálu" superponovaný šum, nastává při průchodu signálu hladinou *trigger*

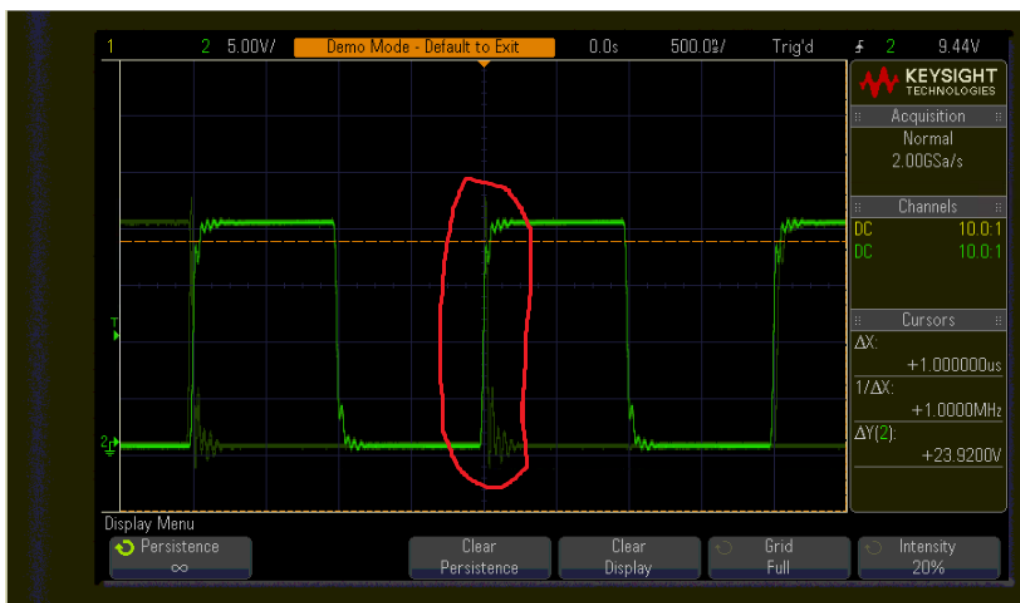


(a) Žádný trigger holdoff – mnoho false triggers

(b) OK trigger holdoff – ignorování šumu

Obrázek 8: Vliv pozdržení triggeru na zarušený signál

level velké množství *trigger conditions*. Z tohoto důvodu triggeruje osciloskop i při "spádné hraně" sinu – ačkoli sinus samotný ryze klesá, šum způsobuje lokální vzestupné hrany vedoucí na trigger patrný na 8a. Na obrázku 8b je již nastaven holdoff polovina periody. Toto samo by bylo špatné nastavení, ale s ohledem na umístění triggeru nad hladinu střední hodnoty je dostatečné, protože signál v oblasti nad triggerem stráví kratší dobu než 500 μ s a holdoff tak zajistí, že sestupná hrana netrigne.



Obrázek 9: Chyba hodinového signálu zobrazená pomocí persistence

Úkol 3 Jednorázové děje

Na obrázku 9 je vykreslen průběh hodinového signálu se zapnutou *persistence* - dříve zobrazené průběhy se nepřepisují a zůstávají vidět, ačkoli se na obrazovku vykreslují i nové zachycené. Pakliže je signál skutečně ideálně periodický, poté budou všechny průběhy dokonale splývat a na displeji bude zobrazena jen jedna tlustá čára. Na obrázku je ale patrná ještě další slabší světle zelená stopa. Jedná se o relikty chyby v signálu, jež nastane třeba jednou za tisíce i miliony period a není tak v lidských silách ji hledat ručně. Persistence nám zajistí, že i takto ojedinělý signál bude zachycen.

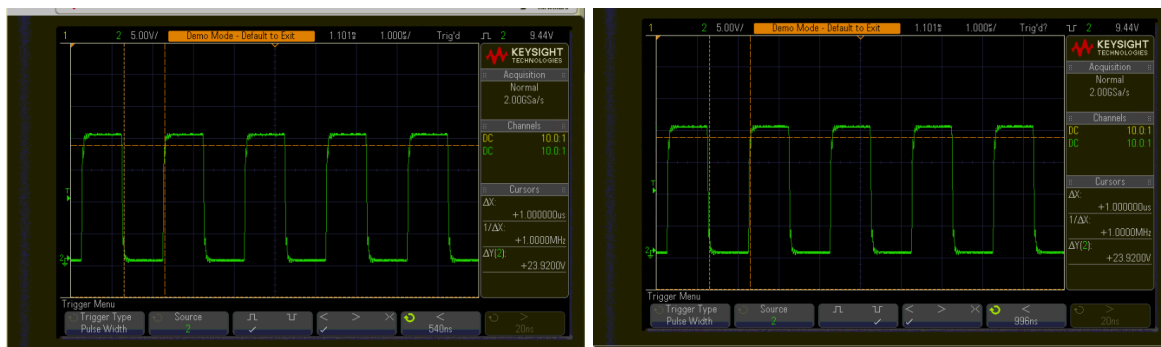
Stejnou chybu je možné detekovat hledáním nesprávné délky pulsu v signálu a nebo aktivací *runt triggeru* (viz obrázek 10). Zde osciloskop detekuje jen pár ns široký kladný puls, který je jistě chybou, neboť šířka pulsů hodin je v řádu mikrosekund. Frekvenční oblast takovéto chyby je pro zajímavost vykreslena na obrázku 18.



Obrázek 10: Identifikace chyby hodin pomocí runt triggeru

Stejná identifikace by se dala provést triggerováním na šířku pulzu a hledáním menší či naopak vyšší hodnoty. Na obrázku 11a je zachycen výstup mého experimentu. Osciloskop úspěšně triggeruje (viz *Trig'd* v levém horním rohu obrazovky) na kladný puls užší než 540 ns. Ovšem kvůli nesprávně nastavenému posunu na časové ose

není glitch vidět, protože je mimo obrazovku... Pro srovnání vedle na obrázku 11b je patrné, že osciloskop není schopen nalézt trigger condition na krátký záporný puls.

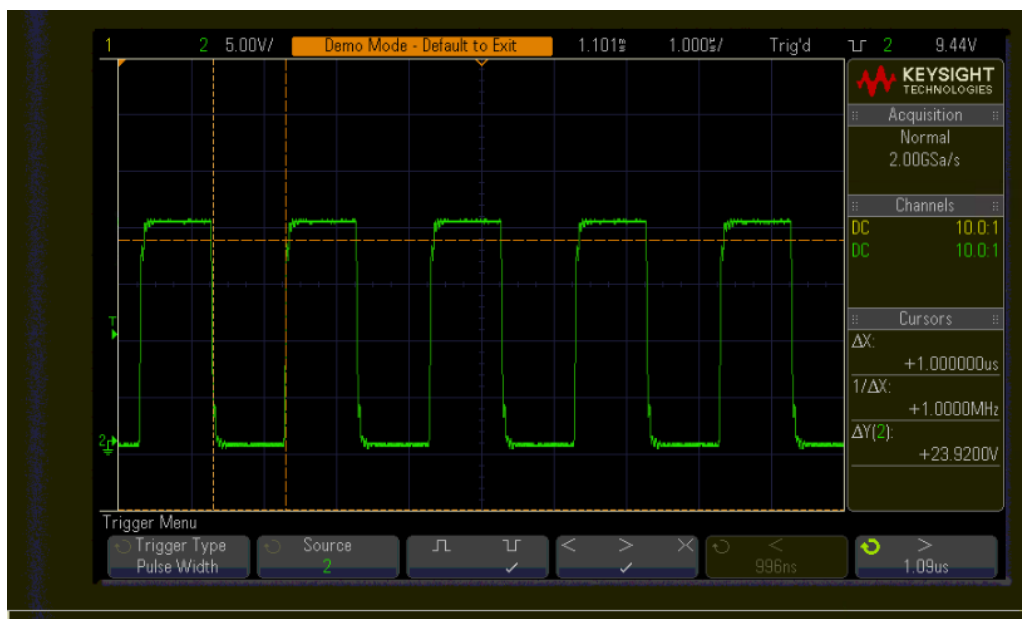


(a) Trigger na malou šířku kladného pulzu

(b) Bez triggeru na malou šířku záporného pulzu

Obrázek 11: Hledání glitche pomocí triggeru nastaveného na malou šířku pulsu

Toto souhlasí s časovým průběhem chyby podle obrázku 10. Všechny zdravé pulsy jsou široké 1 μ s. Při chybě nastane dlouhá nula, krátký kladný puls, opět dlouhá nula a následují zdravé pulsy. Nikde není obsažen záporný puls. Abychom triggerovali na něj, bylo by nejspíše potřeba umístit trigger level kamsi na úroveň zákmitů bezprostředně po kladném pulzu. Podle této logiky by mělo být možné triggerovat na dlouhý záporný puls, neboť ta "dlouhá nula před pulsem" zmiňovaná výše je v principu záporným pulsem. Obrázek 12 sice rovněž trpí mojí chybou se špatnou pozicí na časové ose, potvrzuje ovšem domněnku, že lze triggernout na dlouhý záporný puls.

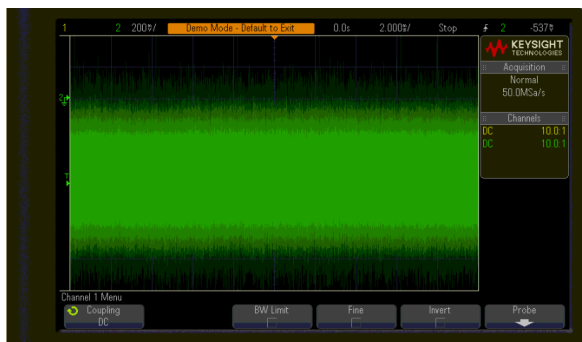


Obrázek 12: Trigger na dlouhý záporný puls

Úkol 4 Vzorkování

Na obrázku 13a je vidět zelený mrak divokého (a na první pohled člověk tuší, že asi i podvzorkovaného) RF signálu. Osciloskop oznamuje, že pro danou časovou základnu 2ms vzorkuje frekvencí $f_s = 50$ MSa/s, čemuž odpovídá vzorkovací krok $h_s = 20$ ns. Protože osciloskop je schopen signál přiblížit na takovou časovou základnu, je možné se podívat přímo na jednotlivé vzorky, jak byly osciloskopem zachytávány. Obrázek 13b ukazuje jednotlivé vzorky (zlomy časového průběhu) zarovnané na dílky časové základny. Kmitočet vzorkování není dostatečný.

Obrázky 14a a 14b napravují tento nedostatek. Zvolením nižší časové základny pouhých 50 μ s se zdánlivě nic moc nezmění při pohledu z dálky (obrázek 14a), ovšem díky čtyřicetkrát vyšší vzorkovací frekvenci 2 GSa/s



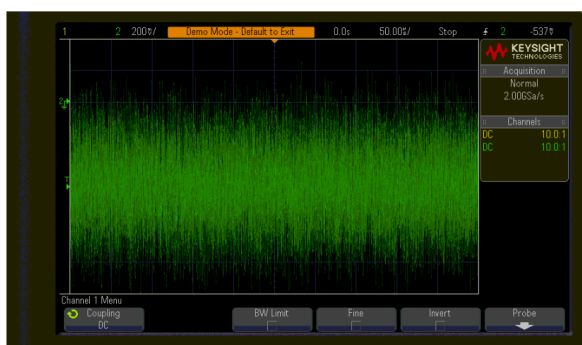
(a) RF signál s časovou základnou 2 ms



(b) RF signál s časovou základnou 20 ns

Obrázek 13: Vysokofrekvenční signál osekáný konečnou vzorkovací frekvencí

není ani při maximálním podporovaném přiblížení na časovou základnu 5ns možné pozorovat ostré zuby způsobené vzorkováním (chyba způsobená vzorkovacím krokem se ztratí ve srovnání s kvantizační chybou převodu analogového signálu AD převodníkem a chybou rozlišení obrazovky).



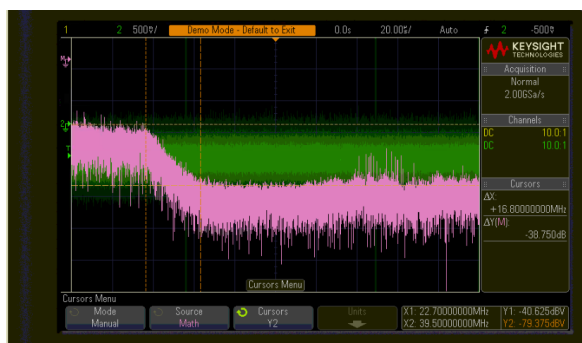
(a) RF signál s časovou základnou 50 μs



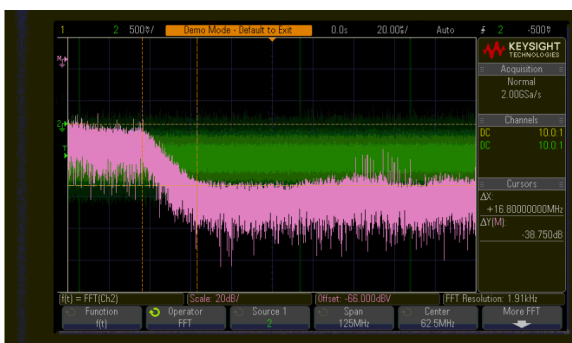
(b) RF signál s časovou základnou 5 ns

Obrázek 14: Zanedbatelnost chyby způsobené konečností vzorkovací frekvence

Úkol 5 Fourierova transformace



(a) Významné hodnoty transformace



(b) Použité nastavení transformace

Obrázek 15: Fourierka RF signálu na plném rozsahu

Navazuji na práci s RF signálem z minulé úlohy. Analýzou průběhu na 14b se dá velmi hrubě odhadnout, že RF signál obsahuje i frekvenci 20 MHz. Podle toho je jasné, že bude potřeba mít velký rozsah FFT. Protože osciloskop fixuje počet vzorků pro výpočet FFT, je velký rozsah spojen s malým rozlišením (= velkým krokem) ve frekvenční oblasti. Pro začátek tedy nastavme FFT na rozsah cca 100 MHz. Výstup je zachycen na obrázku 15a a konfigurace FFT na obrázku 15b. Pomocí kurzorů jsou vyznačeny dvě části spektra. Terminologií teorie signálů se dá hovořit o (alespoň ve srovnání se zbytkem) propustné části pro frekvence od DC až po 22 MHz,

nepropustné (zatlumené) části od frekvence 40 MHz dál a pásnu přechodu na intervalu frekvencí $f \in [22, 40]$ MHz, kde nastává pokles amplitud o 40db na oktávu.

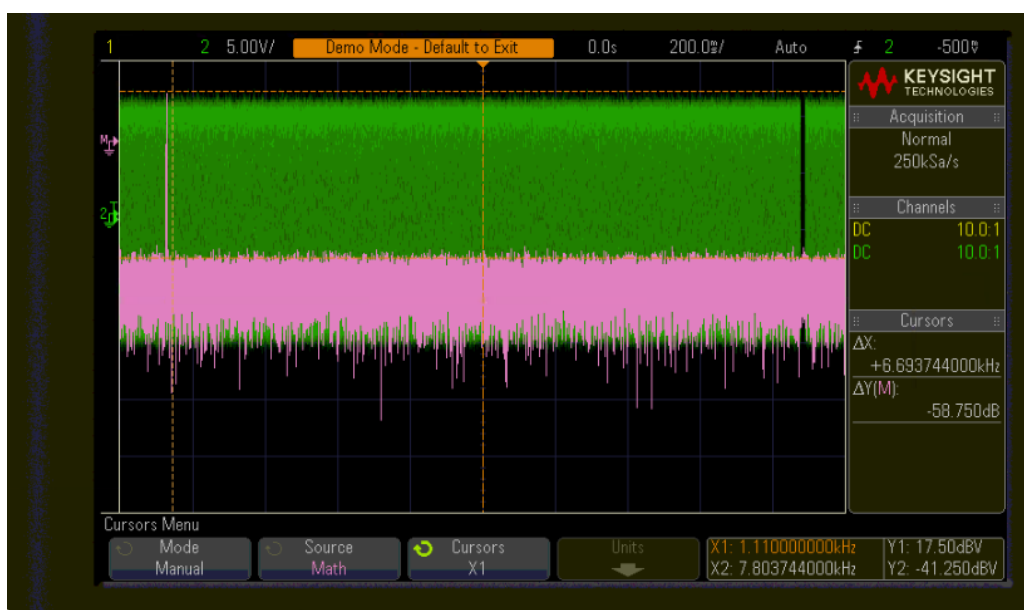
Odhadněme, že bandwidth systému generujícího tento signál je 22 MHz. Vysoké frekvence jsou stoprocentně pouze šumy. Na obrázku 16 je vykreslena fourierka téhož signálu, ovšem s upraveným nastavením - omezením se na menší rozsah frekvencí při zachování počtu vzorků lze dosáhnout lepšího rozlišení (menšího kroku) transformace. Je vidět, že žádná frekvence výrazně nedominuje; RF signál tak lze považovat za bílý šum.



Obrázek 16: Fourierka RF signálu na menším rozsahu

Úkol 5.1 FFT zarušeného signálu

Zajímavá je frekvenční oblast signálu z obrázku 8b vykreslená na obrázku 17. Jedná se o sinus s amplitudou řádově 10V na frekvenci 1kHz zarušený vysokofrekvenčním šumem. Na výstupu FFT je skutečně vidět peak na frekvenci 1kHz s amplitudou 17 dB (+20 dB v amplitudě odpovídá 10 jednotkám v lineární stupnici, takže toto měření velmi obstojně souhlasí s očekáváním), zatímco zbytek spektra je naplněn šumem s amplitudou -40dB.



Obrázek 17: Frekvenční spektrum zarušeného sinu

Úkol 5.2 FFT glitche hodin

V úloze výše byl analyzován hodinový signál na 500 kHz s občasným glitchem. Pro zajímavost je přiloženo jeho frekvenční spektrum na obrázku 18. V levé části je vidět standardní fourierka obdélníkového signálu sestávající z lichých harmonických odeznívajících s rostoucím pořadovým číslem do nuly. Výrazně dál ve spektru je druhý peak příslušící nejspíše k rychlým hranám glitche.



Obrázek 18: Chyba hodin ve frekvenční oblasti

Reference

- [1] Tým vyučujících SME, Zadání úlohy na Moodle, https://moodle.fel.cvut.cz/pluginfile.php/296878/mod_resource/content/1/advanced_remote_scope_v2.pdf
- [2] Keysight academy: Oscilloscope mastery in less than five years, <https://www.youtube.com/watch?v=dQw4w9WgXcQ>
- [3] Motivační hudba, *Kirby dream land theme song* https://www.youtube.com/watch?v=3CS93CdMy_E