**Členové týmu**

* Vít Jánoš (zodpovědný za obsluhu sedmisegmentového displeje)
* Vojtěch Kudela (zodpovědný za obsluhu maticové klávesnice a správu GitHub)
* David Matějček (zodpovědný za komparátor a testování)
* Antonín Putala (zodpovědný za generátory, PWM modulátor a demo nahrávky)

**Teoretický popis a vysvětlení**

**Zvuk** je podélné mechanické vlnění. Zvuková vlna působí výchylku od atmosférického tlaku. Aby byl zvuk pro člověka slyšitelný, musí být o kmitočtu 20 Hz až 20 kHz a větší výchylku než práh slyšení *P*0 = 2·10-5 Pa. Dlužno podotknout, že citlivost lidského sluchu je frekvenčně závislé, práh slyšení i kmitočtový rozsah se liší dle jednotlivce. Lidské ucho vnímá hlasitost logaritmicky, proto se hlasitost vyjadřuje v decibelové míře.

*L*P je hladina akustického tlaku.

Periodické zvukové signály se nazývají **tóny**. Tyto tóny mají svou základní frekvenci, která je označovaná jako výška. Podle výšky také nese tón název. V hudební teorii se používá označení oktáva pro zdvojnásobení kmitočtu. Pokud zdvojnásobíme kmitočet tónu, ponese stejný název pouze bude ležet ve vyšší oktávě. Jako základní tón se užívá komorní A s kmitočtem *f* = 440 Hz. Oktáva se v evropské hudební teorii rozděluje na 12 dílů. Ty se nazývají půltóny. Každému půltónu je přiřazen kmitočet. Tento kmitočet se určí jako součin základního kmitočtu a poměru pro konkrétní půltón. Pokud změníme frekvenci základního tónu, změní se kmitočty všech půltónů.

Princip, podle kterého jsou přiřazovány poměry k půltónům nazýváme **ladění**. V minulosti byl užívány různé typy ladění, které měly své výhody i neduhy (např.: Pythagorejské, Středotónové, Parejovo…). V současném době se používá téměř výhradně rovnoměrně temperované ladění. V rovnoměrně temperovaném ladění je poměr mezi dvěma po sobě jdoucími půltóny .

Z tónu A se tedy získá A# vynásobením .

A tímto způsobem by bylo pokračováno pro zbývající půltóny.

**Tab. 1 Tabulka kmitočtů jednotlivých tónů**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Název | Kmitočet [Hz] | Název | Kmitočet [Hz] | Název | Kmitočet [Hz] |
| A | 440,0 | C# | 554,4 | F | 698,5 |
| A# | 466,2 | D | 587,3 | F# | 740,0 |
| B\* | 493,9 | D# | 622,3 | G | 784,0 |
| C | 523,3 | E | 659,3 | G# | 830,6 |

\* V českých a německých zemích je zvykem nazývat tento tón H. Oproti tomu jako B je označován tón A#. Zde je uváděné pojmenování obvyklé v anglosaských zemích.

Bylo by samozřejmě možné pokračovat dál, nicméně následoval by kmitočet přesně dvojnásobný oproti základnímu tónu, tedy bylo dosaženo celé oktávy.

Hudební tón není tvořen pouze základním kmitočtem, nýbrž též vyššími harmonickými složkami (alikvótními tóny), což jsou celistvé násobky základního kmitočtu. Tóny stejné výšky s odlišným obsahem vyšších harmonických složek se liší co do **barvy tónu**. To lze demonstrovat na rozdílném zvuku jednotlivých hudebních nástrojů. Vhodnou kombinací vyšších harmonických lze věrně napodobit libovolný hudební nástroj. Odlišná barva tónu je demonstrována na třech základních průbězích: obdélníkovém, trojúhelníkovém a pilovém.

Generace zvukových signálů pomocí FPGA působí problém, protože zvukový signál je spojitý, zatímco FPGA má pouze dvoustavové výstupy. Řešením je užití **pulzně šířkové modulace** (PWM). PWM je modulace, při které se střída výstupního obdélníkového signálu mění v závislosti na okamžité hodnotě modulovaného signálu. Velká výchylka okamžitého signálu se projeví větší střídou, menší výchylka menší střídou. Hlavním benefitem je 1bitový výstup, který může být přímo zaveden do reproduktoru, Pokud je potřeba získat původní signál stačí užít dolní propust.

PWM modulátor je v principu komparátor. Jako referenční průběh je nutné použít pilový průběh. Je nutné, aby byl kmitočet referenčního byl alespoň o řád větší než nejvyšší kmitočet modulačního signálu. Aby zařízení bezchybně fungovalo je nutné, aby měl v každém čase referenční průběh větší maximální hodnotu a menší minimální hodnotu než modulační signál. Jinak by měl výstup střídu 100 %, respektive 0 %, což není možno. Kmitočet použitého je pouhých 10 kHz, což je nicméně v této konkrétní aplikaci postačující, avšak nebylo by, pakliže by byl rozsah rozšířen směrem k vyšším tónům.

Obsah obrázku text, diagram, řada/pruh, Písmo

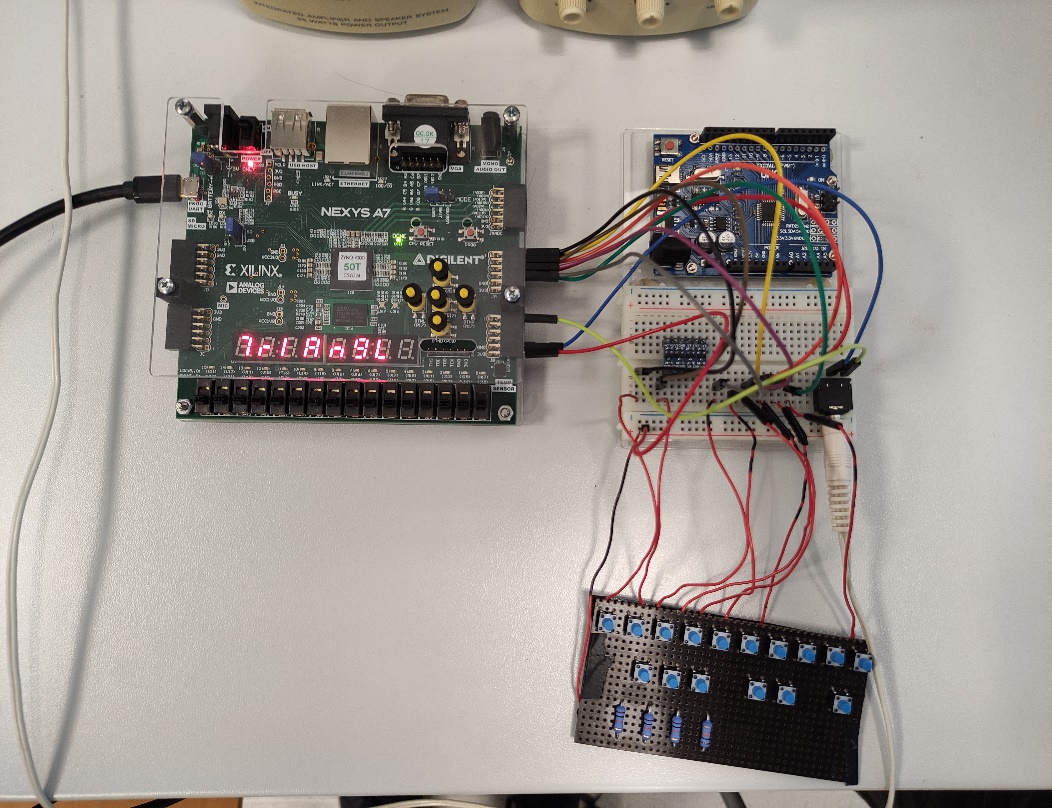
Popis byl vytvořen automaticky

**Obr. 1 Princip PWM modulátoru – převzato z 2**

Je pochopitelné, že modulační i referenční průběh je nutné uvnitř FPGA vyjádřit pomocí číslicových hodnot. Signál je vyjádřen vzorky, jímž je přiřazena 8bitová číselná hodnota. Protože tyto průběhy mohou nabývat též záporných hodnot, je hodnota 127 (x“7F“) užívána coby nulová úroveň, vyšší hodnoty jsou kladné, nižší záporné.

**Hardwarový popis a demo aplikace**

Zařízení bylo oživeno a testováno na desce [NEXY-A7-50T](https://digilent.com/reference/programmable-logic/nexys-a7/start). Tato deska obsahuje mimo jiné osmimístný sedmisegmentový display, 16 LED diod a 5 tlačítek, což jsou periferie, které byly užity. Další zařízení byla připojena na vnější porty JA a JB. Na ty byly připojeny vnější periférie Jack 3,5mm pro připojení reproduktoru a maticová klávesnice upravená jako klaviatura piana pro zadávání tónu do zařízení.



**Obr. 2 Náhled na zařízení**

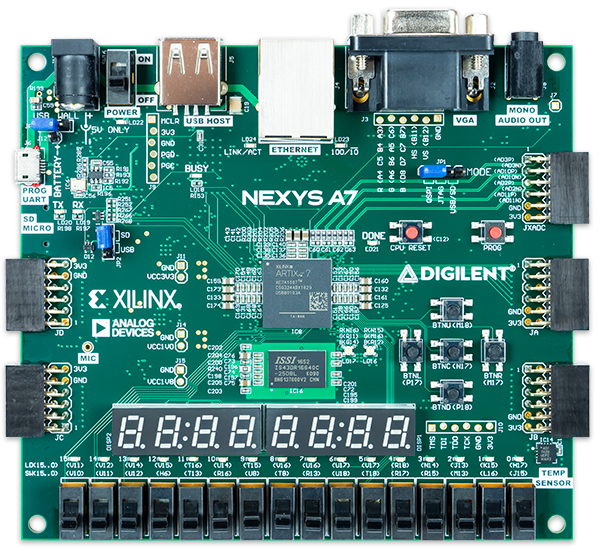
[**TOP LEVEL**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/top_level.vhd)

***Obsah obrázku text, diagram, snímek obrazovky, číslo

Popis byl vytvořen automaticky***

**Obr. 3** [**TOP\_LEVEL**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/top_level.vhd) **zařízení**

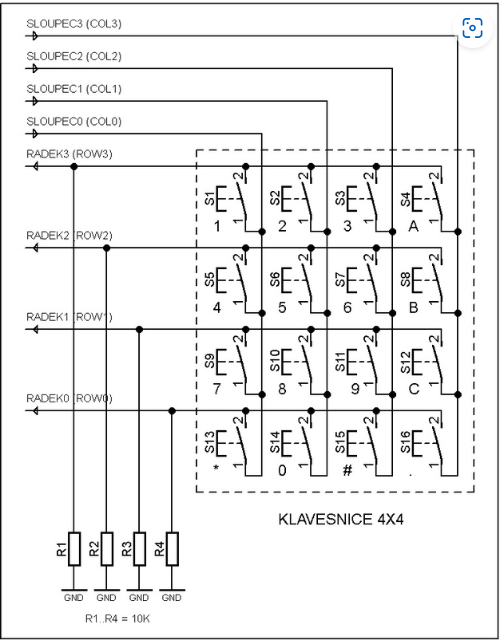
Blok [TOP\_LEVEL](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/top_level.vhd) je nejvyšší úrovní celého zařízení. Zde jsou utvořeny vývody pro jednotlivé periférie, které jsou poté přiřazeny pomocí [contrainu](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/constrs_1/new/nexy-a7-50t.xdc). Vývody BTNC, BTND, BTNU, BTNR a BTNL jsou připojeny k jednotlivým tlačítkám na desce. Vývod LED[15:0] je připojen k LED diodám na desce. Vývody CA, CB, CC, CD, CE, CF, CG, DP, AN[7:0] řídí sedmisegmentový display. Vývod CLK100MHZ potom je připojen na zdroj hodinových pulzů a AUD\_PWM je připojen přes Sallen-Key dolní propust na mono Jack. Vývody s0, s1, s2, s3 jsou připojeny k řádkům maticové klávesnice, vývody r0, r1, r2, r3 jsou připojeny k řádkům. Vývod JB je připojen na vnější Jack, který slouží k připojení reproduktorů nebo sluchátek. Konkrétně je připojen na pin 1 na svorkovnici JB.



**Obr. 4 Deska** [**NEXY-A7-50T**](https://digilent.com/reference/programmable-logic/nexys-a7/start)

**Maticová klávesnice**

Připojovat tlačítka na jednotlivé porty FPGA je vysoce nehospodárné. Pro zvýšení efektivity využití portů je vhodné využít maticové zapojení spínačů. Tlačítka jsou zapojeny do řádků a sloupců, které jsou vyvedeny. Při stisknutí tlačítka dojde k propojení řádku a sloupce, čímž je možné identifikovat stisknutí tlačítka. Zařízení využívá maticovou klávesnici 4x4. K připojení postačuje 8 portů. Pokud by tlačítka byla připojena samostatně, bylo by portů potřeba dvojnásob a je zřejmé, že s přibývajícím počtem řádků a sloupců efektivita roste.



**Obr. 5 Maticová klávesnice ‒ zapojení (převzato z** [**11**](https://eluc.ikap.cz/verejne/lekce/924)**)**

Použité pull-up rezistory mají hodnotu 10 kΩ. Deska byla konstruovaná na více účelové desce bez měděných plošek, tedy montáž byla poměrně náročná a s nízkou jakostí, avšak pro potřeby otestování zařízení zcela postačující. Tlačítka byla seřazena tak, aby to co možná nejvíce zjednodušilo montáž, v jejich pořadí je tedy pomíchané. Jednotlivé vodiče jsou odděleny značkou vytvořenou černým fixem. Je na nich vyznačen žádný, jeden, dva nebo tři černé pruhy. Vodiče bez návlečky jsou vyvedené sloupce, vodiče s návlečkou jsou vyvedené sloupce. Vodič označený dlouhým černým pruhem slouží k připojení napájecího napětí.

Obsah obrázku snímek obrazovky

Popis byl vytvořen automaticky

**Obr. 6 Připojení jednotlivých vodičů na svorkovnici JA**

Obsah obrázku elektronika, Elektronické inženýrství, Obvodoví součástka, Elektrické vedení

Popis byl vytvořen automaticky

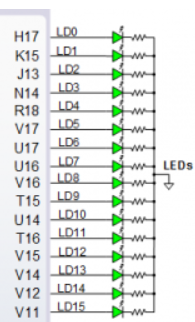
**Obr. 7 Náhled na použitou maticovou klávesnici**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 3, 1 5 | |  |  | 3, 3 6 | | 3, 2 7 | |  |  | 3, 0 4 | | 1, 1 9 | | 1, 0 8 | |  |  |  |  | sloupec, řádek kód po přečtení |  |
|  | 2, 0 0 | | 2, 1 1 | | 2, 3 2 | | 2, 2 3 | | 1, 2 B | | 1, 3 A | | 0, 3 E | | 0, 2 F | | 0, 1 D | | 0, 1 C | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Obr. 8 Tabulka rozložení tlačítek na maticovém displeji ‒ řádek, sloupec, kód, který zařízení vygeneruje po stisku tlačítka**

**Sedmisegmentový displej a LED**

Při práci se sedmisegmentovým displejem i LED diodami je nutné myslet na to, zda mají aktivní úroveň ve vysoké nebo nízké úrovni. U LED diod je situace poněkud jednodušší, neboť aktivní jsou ve vysoké úrovni. To lze zjistit ze schématu.



**Obr. 9 Zapojení LED diod na desce NEXY-A7-50T**

Opačná situace je u sedmisegmentového displeje. Zde je aktivní nízká úroveň. Jedná se o zapojení se společnou anodou. Anody jednotlivých sedmisegmentových displejů jsou vyvedeny a umožňují vypínat a zapínat jednotlivé pole displeje. Pro zapnutí displeje je rovněž nutné přivést nízkou úroveň, neboť jednotlivá pole displeje jsou spínána přes bipolární tranzistory typu PNP. Jednotlivé segmenty jsou rovněž vyvedeny, nicméně jsou propojeny s vývody ostatních polích. Toto uspořádání je vhodné pro dynamické řízení displeje.

Obsah obrázku text, diagram, Písmo, řada/pruh

Popis byl vytvořen automaticky

**Obr. 10 Zapojení sedmisegmentového displeje na desce NEXY-A7-50T**

**Softwarový popis**

Obsah obrázku text, diagram, Plán, schématické

Popis byl vytvořen automaticky

**Obr. 11 Schéma zapojení DISPLAY\_AND\_GEN\_AND\_DEMO**

Celé zařízení je možno si rozdělit do tří velkých bloků: [DISPLAY\_AND\_GEN](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/display_and_gen.vhd), [DEMO\_MAIN](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_main.vhd) a [MATRIX\_CONTROL](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/matrix_control.vhd). Každý z nich obsluhuje odlišnou část zařízení. [DISPLAY\_AND\_GEN](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/display_and_gen.vhd) má podřízenou obsluhu displeje, generaci signálů, PWM modulátor a obsluhu tlačítek volume a mode. Obsahuje mimo jiné blok MAIN\_LOOP, který umožňuje přechod zařízení mezi jednotlivými módy. Blok [DEMO\_MAIN](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_main.vhd) obsahuje informace o frekvenci a hlasitosti tří demo nahrávek a pomocnou logiku pro vytvoření příslušného signálu. [MATRIX\_CONTROL](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/matrix_control.vhd) má za úkol obsluhovat maticovou klávesnici a předávat ostatním blokům informaci o tom, které tlačítko bylo stisknuto. Obvod [CODER\_1\_FROM\_16](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/coder_1_from_16.vhd) je pomocný blok, který zajišťuje rozsvícení odpovídající diody, vždy při stisku klávesy. O něm budou bližší informace c souvislosti s blokem [MATRIX\_CONTROL](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/matrix_control.vhd). Všechny tyto bloky jsou zastřešeny design sourcem [DISPLAY\_AND\_GEN\_AND\_DEMO](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/display_and_gen_and_demo.vhd), který je přímo podřízen TOP\_LEVELU.

1. **Blok** [**DISPLAY\_AND\_GEN**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/display_and_gen.vhd)

Obsah obrázku text, diagram, řada/pruh, Vykreslený graf

Popis byl vytvořen automaticky

**Obr. 12 Schéma zapojení DISPLAY\_AND\_GEN**

Blok DISPLAY\_AND\_GEN zastřešuje dva velké bloky SOUND\_GENERATOR a DISP\_CON Mimo to obsahuje logiku, která nastavuje reset komparátoru. Komparátor je resetován, pokud není nic stisknuto a zařízení nehraje demo nahrávku nebo pokud je reset komparátoru nařízen blokem DISP\_CON.

**1.1. Blok** [**SOUND\_GENERATOR**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/sound_generator.vhd)

Úkolem SOUND\_GENERATOR je generovat PWM modulovaný signál, který obsahuje zvukovou informaci. Tento signál je buďto vygenerován uvnitř tohoto bloku nebo brán zvenčí vstupy pro dema.

Obsah obrázku diagram, text, Plán, Technický výkres

Popis byl vytvořen automaticky

**Obr. 13 Schéma zapojení** [**SOUND\_GENERATOR**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/sound_generator.vhd)

Blok obsahuje tři generátory, které generují jednotlivé průběhy, [osmikanálový multiplexer](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/mux3bit.vhd), převodník [boot\_to\_freq](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/boot_to_freq.vhd), generátor pulzů [clock\_enable](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/clock_enable.vhd) a samotný [PWM modulátor](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/PWM_mod.vhd). [**Osmikanálový multiplexer**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/mux3bit.vhd) umožňuje přepínání vstup PWM modulátoru mezi třemi střídavými průběhy a třemi demo nahrávkami. Je adresován signálem mode. 4bitový signál freq obsahuje informaci o frekvenci, nicméně v této podobě s ním nemohou generátory pracovat, proto je tato hodnota [**převodníkem boot\_to\_freq**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/boot_to_freq.vhd) převedena na číselnou hodnotu, která odpovídá desetinásobku frekvence v Herzích. Protože tlačítka maticové klávesnice nejdou po sobě, bylo nutné zajistit jejich uspořádání softwarově, a to plní tento převodník. Převádí 4bitovou vstupní hodnotu na 16bitový výstup dle následující tabulky.

**Tab. 2 Převodní tabulka boot\_to\_freq**

|  |  |  |
| --- | --- | --- |
| **inpu (hex)** | **output** | **odpovídá** |
| 0 | x"0898" | A |
| 1 | x"09A5" | B |
| 2 | x"0A38" | C1 |
| 3 | x"0B78" | D1 |
| 4 | x"0E74" | F#1 |
| 5 | x"091B" | A# |
| 6 | x"0AD4" | C#1 |
| 7 | x"0C27" | D#1 |
| 8 | x"1235" | A#1 |
| 9 | x"1039" | G#1 |
| A | x"0DA4" | F1 |
| B | x"0CE0" | E1 |
| C | x"1470" | C2 |
| D | x"134B" | B1 |
| E | x"0F50" | G1 |
| F | x"1130" | A1 |

**1.1.1.** [**Osmikanálový multiplexer**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/mux3bit.vhd)

Design source nese jméno mux3bit podle počtu bitů adresového vstupu. Multiplexer je elektronický přepínač, který v závislosti na adresových vstupech připojí k výstupu vždy pouze jeden vstup. Použitá rutina umožňuje pracovat s N-bitovými vstupy. Protože je pracováno s osmibitovým signálem, je i užitý multiplexer osmibitový. Blok realizuje funkci popsanou tabulkou:

**Tab. 3 Tabulka popisující funkci osmikanálového multiplexeru**

|  |  |
| --- | --- |
| sel | output |
| "000" | in\_0 |
| "001" | in\_1 |
| "010" | in\_2 |
| "011" | in\_3 |
| "100" | in\_4 |
| "101" | in\_5 |
| "110" | in\_6 |
| "111" | in\_7 |

**1.1.2 Blok** [**PWM\_mod**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/PWM_mod.vhd)

Obsah obrázku text, diagram, řada/pruh, snímek obrazovky

Popis byl vytvořen automaticky

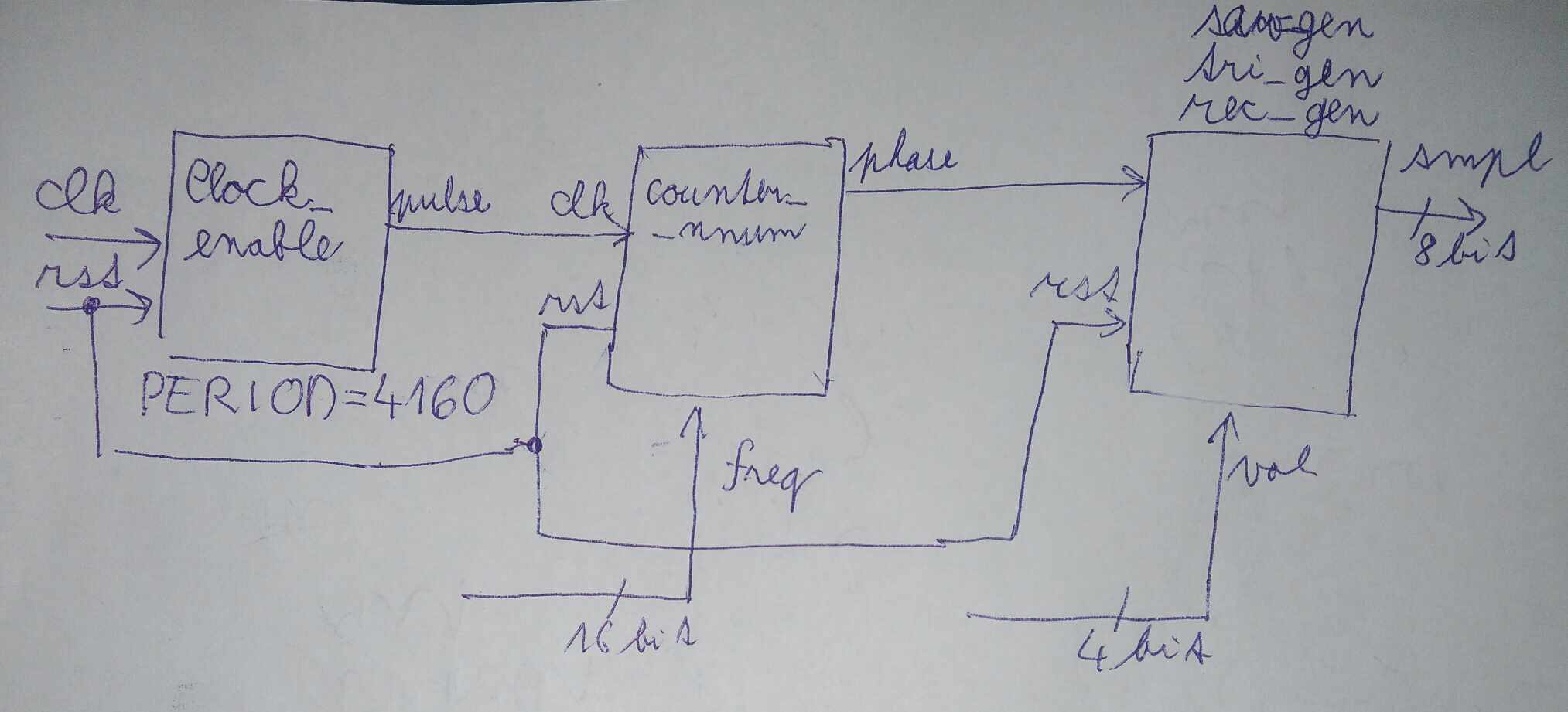
**Obr. 14 Schéma zapojení PWM\_mod**

Samotný PWM generátor se skládá z generátoru pilového průběhu a [komparátoru](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/kompareNbit.vhd). Pilový průběh vytváří bloky [CLOCK\_ENABLE](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/clock_enable.vhd) a [SIMPLE\_COUNTER](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/simple_counter.vhd). Oba bloky byly napsány a otestovány v rámci počítačového cvičení, a tedy zde nejsou podrobně popsány. Pro úplnost je pouze uvedeno, že blok [CLOCK\_ENABLE](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/clock_enable.vhd) jednou za nastavený počet cyklů pulz, který setrvá do příchodu další náběžné hrany. Blok [SIMPLE\_COUNTER](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/simple_counter.vhd) je pak vzestupný binární N-bitový čítač (v tomto případě osmibitový), který inkrementuje svůj stav vždy po příchodu hodinového signálu, je-li jeho vstup en v logické jedničce. Protože je jemu předřazený blok [CLOCK\_ENABLE](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/clock_enable.vhd), bude měnit svůj obsah vždy při příchodu pulzu z tohoto bloku. Poté, co čítač dosáhne maximální hodnoty, přeteče a začne čítat od nuly. Výsledkem je tedy periodicky se opakující nárůst číselné hodnoty, která reprezentuje pilový průběh. Protože [CLOCK\_ENABLE](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/clock_enable.vhd) generuje pulz co 40 hodinových pulzů a čítač čítá do 255 (28 ‒ 1) je frekvence pilového signálu přibližně 10 kHz.

Druhou neméně důležitou částí je komparátor, jenž je obsažen v design sourcu [kompareNbit](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/kompareNbit.vhd). Vzhledem na počet bitů modulačního signálu musí být i komparátor osmibitový. Jedná se o komparátor řízený hodinovým signálem s asynchronním resetem aktivním v nule, který je označen en. Protože zbývající bloky mají reset aktivní v jedničce, je ke společnému resetovacímu vstupu připojen přes invertor. Komparátor má dva vstupy „a“ a „b“. Jako vstup „a“ je připojen referenční pilový průběh, jako vstup „b“ modulační signál. Rutina umožňuje pracovat až se třemi výstupy: menší, větší a rovná se; nicméně pro aplikaci komparátoru coby funkčního bloku PWM modulátoru je bohatě postačující výstup jediný, konkrétně pak byl užit výstup b\_greater.

**1.1.3. Generátory tvarových průběhů**

Generátory tvarových kmitů používají velmi podobná princip, jako se užívá v generátorech na principu přímé digitální syntézy (DDS). Rozdílné je pouze získávání hodnoty vzorku. Zatímco v DDS je hodnota vzorku uložena v paměti a v požadovaný okamžik je pouze vytažena, v zařízení je hodnota vzorku počítána pokaždé znovu.



**Obr. 15 Zapojení generátorů tvarových průběhů**

V paměti DDS je nahraný průběh, ale jakým způsobem měnit jeho frekvenci? V zásadě existují dvě řečení. První je měnit frekvenci hodinových kmitů, tedy jak často jsou vzorky z paměti vytahovány. To klade vysoké nároky na generátor hodinových pulzů a na jeho přeladitelnost, proto se tento princip v praxi neujal. Oproti to se užívá druhé řešení. Generátor hodinových kmitů má stálý kmitočet, mohu tedy využít velmi stabilní krystalový rezonátor, ovšem nevytahuji vzorky postupně, nýbrž vždy několik vzorků přeskočím. Pokud je vhodně zvolena frekvence hodinových pulzů a počet vzorků, například bez přeskakování byl kmitočet 0,1 Hz (jak je tomu též v tomto zařízení), je možné měnit kmitočet právě o tento frekvenční krok. Maximální kmitočet pak je dán Shannon-Kotělnikovým teorémem, v tomto případě bylo pracováno s 120 000 vzorky, takže teoretický maximální dosažitelný kmitočet byl 6 kHz.

Toto přeskakování zajišťuje blok nazvaný [COUNTER\_NNUM](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/counter_nnum.vhd). Název dobře vystihuje činnost tohoto bloku. V principu se jedná o vzestupný čítač, který ale neinkrementuje, ale přičítá ke svému stavu hodnotu, která mu byla přivedena na vstup freq. Aby správně plnil svou roli je třeba zkrátit jeho cyklus, a to ne pouze na hodnotě 199 999, nýbrž je nutno tuto hladinu s měnícím se vstupem freq ponížit právě o tuto hodnotu, aby nedošlo k přetečení.

Čítač dostává čítací pulz jednou za 4160 cyklů. To zajišťuje blok [CLOCK\_ENABLE](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/clock_enable.vhd). Hodnota 4160 cyklů nebyla vypočtena, ale empiricky zjištěna na základě simulací, neboť se ukázalo, že očekávaný kmitočet, neodpovídal teoreticky vypočtenému.

Bloky zabezpečující výpočet hodnoty vzorku jsou [REC\_GEN](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/rec_gen.vhd) pro obdélníkový průběh, [TRI\_GEN](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/tri_gen.vhd) pro trojúhelníkový průběh a [SAW\_GEN](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/saw_gen.vhd) pro pilový průběh. Tyto bloky realizují následující funkce:

Obdélníkový průběh:

Trojúhelníkový průběh:

Pilový průběh:

Proměnná *vol* určuje svou hodnotu podle hodnoty na vstupu vol. Určuje rozkmit generovaného signálu. Hodnota *con\_step* činí 8 a *con\_center* 127. Hodnota *con\_ceil* je 59999. Tyto hodnoty byly určeny s ohledem na 8bitový výstupní signál. Maximální rozkmit signálu činí 240, minimální pak 16.

* 1. **Blok** [**DISP\_CON**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/disp_control.vhd)

Tento blok se zabývá ovládáním osmi 7-mi segmentových displejů. Má osm programovatelných stavů. Které se přepínají změnou módů z maniloopu. Kdy 6 stavů je pevně určeno, 1 stav je nevyužit a 1 stav je závisí na nastavení hlasitosti.

Blok se skládá: z 8 multiplexerů řízených externě mainloopem, z 1 multiplexeru, který je interně řízen hodinovým signálem přes [CLOCK\_ENABLE](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/clock_enable.vhd).

Obsah obrázku text, diagram, řada/pruh, Paralelní

Popis byl vytvořen automaticky

**Obr. 16 Schéma** [**DISP\_CONTROL**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/disp_control.vhd)

* + 1. [**MUX**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/mux/mux.vhd) **\_**

Jedná se o 3-bitový, multiplexer má tedy 8 vstupů, u kterých je možno nastavit počet bitů, a jednoho výstupu. Podle stavu na ovládacích vstupech přiřadí určitý vstup na výstup.

Obsah obrázku text, snímek obrazovky, číslo, diagram

Popis byl vytvořen automaticky

**Obr. 17 Schéma multiplexeru** [**MUX**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/mux/mux.vhd)

* + 1. [**Mux0-7**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/mux/mux.vhd)

Zajišťují řízení prvního 7-mi segmentového displeje a na jeho vstupy je přiváděn 5-bitový kód, který odpovídá požadovanému symbolu pro každý z módů, který je určen stavem ovládacích vstupů. Kdy výstupy jsou přiváděny na [main\_mux](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/mux/mux3bit.vhd), kde jsou dále zpracovány.

* + 1. [**Main\_mux**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/mux/mux3bit.vhd)

Jeho úkolem je přepínat přiváděný kód na osm 7-segmentových displejů. Každé 2 ms. přepne zobrazovaný symbol, podle toho, který displej je aktivní. Jeho ovládací vstupy jsou řízeny pomocí [simple\_counter](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/simple_counter.vhd).

* + 1. [**Simple\_counter**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/simple_counter.vhd)

Zprostředkovává synchronizaci multiplexeru [main\_mux](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/mux/mux3bit.vhd) a převodníku one\_to\_eight. Toho docílí tak, že při každém impulsu na clock, který dorazí z čítače [clock\_enable](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/clock_enable.vhd), zvýší hodnotu výstupu o jedničku.

Obsah obrázku text, diagram, snímek obrazovky, řada/pruh

Popis byl vytvořen automaticky

**Obr. 18schéma Simple\_counter**

* + 1. [**One\_to\_eight**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/prev.vhd)

Tento převodník slouží k přepínání anod jednotlivých displejů na hodnotu LOW, a tím displej odblokuje. K tomuto přepínání dochází každé 2 ms.

|  |  |
| --- | --- |
| one\_to\_eight | |
| vstup | výstup |
| “000“ | x“7F“ |
| “001“ | x“BF“ |
| “010“ | x“DF“ |
| “011“ | x“EF“ |
| “100“ | x“F7“ |
| “101“ | x“FB“ |
| “110“ | x“FD“ |
| “111“ | x“FE“ |
| others | x“FF“ |

**Obr. 19.převodní tabulka** [**one\_to\_eight**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/prev.vhd)

Obsah obrázku text, diagram, řada/pruh, Plán

Popis byl vytvořen automaticky

**Obr. 20-schéma převodníku** [**one\_to\_eight**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/prev.vhd)

* + 1. [**Clock\_enable**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/clock_enable.vhd)

Tento čítač čítá náběžné hrany hodinového signálu, kdy při dosažení přednastavené hodnoty přivádí vysokou úroveň na jeho výstup po dobu trvání jedné periody hodinového signálu, poté se čítač resetuje.

Obsah obrázku diagram, text, řada/pruh, Plán

Popis byl vytvořen automaticky

**Obr. 21-čítač** [**Clock\_enable**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/clock_enable.vhd)

* + 1. [**Display**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/bin2seg.vhd)

Jedná se o převodník binárního kódu na segmenty displeje. Obsahuje I vstup Clear , který při přivedení vysoké úrovně zhasne všechny segmenty displeje.

|  |  |  |  |
| --- | --- | --- | --- |
| input | output (hex) | input | output (hex) |
| "00000" | x"01" | "10000" | x"48" |
| "00001" | x"4F" | "10001" | x"79" |
| "00010" | x"12" | "10010" | x"49" |
| "00011" | x"06" | "10011" | x"71" |
| "00100" | x"4C" | "10100" | x"68" |
| "00101" | x"24" | "10101" | x"4A" |
| "00110" | x"20" | "10110" | x"6A" |
| "00111" | x"0F" | "10111" | x"18" |
| "01000" | x"00" | "11000" | x"7A" |
| "01001" | x"04" | "11001" | x"0F" |
| "01010" | x"08" | "11010" | x"41" |
| "01011" | x"60" | "11011" | x"6C" |
| "01100" | x"31" | "11100" | x"77" |
| "01101" | x"42" | "11101" | x"78" |
| "01110" | x"30" | "11110" | x"7F" |
| "01111" | x"38" | "11111" | X |

**Obr. 22-Převodní tabulka** [**Display**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/bin2seg.vhd)

Obsah obrázku text, diagram, řada/pruh, snímek obrazovky

Popis byl vytvořen automaticky

**Obr. 23-schéma převodníku** [**Display**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/bin2seg.vhd)

* + 1. [**Main loop**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/main_loop.vhd)

Tento stavový automat má z úkol nastavovat módy a hlasitost podle inputu uživatele pomocí tlačítek. Dvě pro změnu módu a dvě pro nastavení hlasitosti. Má 7 normálních stavů a 1 přechodný, který slouží jako ochrana, kdyby se dostal do nějakého mezi stavu a zároveň slouží jako přechod z volume do ostatních stavů.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| State | mux\_mode | mux\_let | clear | rst\_komp | rst\_play |
| SAFE | mode | "000" | '1' | '1' | '1' |
| SQUARE | "000" | "001" | '0' | '0' | '1' |
| TRIANGLE | "001" | "010" | '0' | '0' | '1' |
| SAW | "010" | "011" | '0' | '0' | '1' |
| DEMOA | "011" | "100" | '0' | '0' | '0' |
| DEMOB | "100" | "101" | '0' | '0' | '0' |
| DEMOC | "101" | "110" | '0' | '0' | '0' |
| VOLUME | mode | "111" | '0' | '1' | '1' |

**Obr. 24-Převodní tabulka** [**main\_loop**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/main_loop.vhd)

Obsah obrázku diagram, Plán, schématické, řada/pruh

Popis byl vytvořen automaticky

**Obr. 25Schéma** [**main\_loop**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/main_loop.vhd)

* + 1. [**Volume control**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/volume_control.vhd)

Skládá se ze 3 bloků, a to [debounce](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/debounce.vhd) a [up\_down\_counter\_limited](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/updown_counter_limited.vhd) a jednoho logického hradla OR. Hlavním účelem tohoto bloku je sledovat tlačítka na měnění hlasitosti, pamatovat si úroveň hlasitosti a upozornit [main\_loop](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/main_loop.vhd), že došlo ke změně hlasitosti. Counter jedním tlačítkem počítá nahoru a druhým dolů, je limitován od 0 do 15.

**Obsah obrázku text, diagram, snímek obrazovky, číslo

Popis byl vytvořen automaticky**

**Obr. 26-Schéma** [**volume\_control**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/volume_control.vhd)

* + 1. [**Mode control**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/mode_control.vhd)

Skládá se ze 3 bloků, a to [debounce](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/debounce.vhd) a [up\_down\_counter](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/updown_counter.vhd) a jednoho logického hradla OR. Hlavním účelem tohoto bloku je sledovat tlačítka na měnění módu, pamatovat si právě zvolený mód a upozornit [main\_loop](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/main_loop.vhd), že došlo ke změně módu. Counter jedním tlačítkem počítá nahoru a druhým dolů. Není limitován, při překročení maximální hodnoty počítá znova.

*Obsah obrázku text, diagram, Plán, číslo

Popis byl vytvořen automaticky*

**Obr. 27-Schéma** [**mode\_control**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/OldRoutine/Complete/mode_control.vhd)

1. **Blok** [**DEMO\_MAIN**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_main.vhd)

Tato část je jakýmsi pomyslným rozšířením bloku [SOUND\_GENERATOR](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/sound_generator.vhd), využívá stejné rutiny a pouze plynule v čase mění vstupní parametry. Zařízení umožňuje zahrát celkem 3 melodie. Jedná se o skladbu [Für Elise](https://www.youtube.com/watch?v=q9bU12gXUyM) od L. van Beethovena (DEMOA), [Do you heard the people sing?](https://www.youtube.com/watch?v=gYb9sRLUDyM) od Claude-Michel Schönberga (DEMOB) a skladbu [Entertainer](https://www.youtube.com/watch?v=Fxk9qwCFf8s) od S. Joplina (DEMOC). Pro zjednodušení budou tyto skladby nazývány pouze DEMO a odpovídající písmeno.

Obsah obrázku diagram, Plán, řada/pruh, schématické

Popis byl vytvořen automaticky

**Obr. 28 Schéma zapojení** [**DEMO\_MAIN**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_main.vhd)

Základem jsou tři bloky, z nichž každý obsahuje informaci o jedné nahrávce: [DEMO\_INS](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_ins.vhd), [DEMO\_INS\_1](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_ins_1.vhd) a [DEMO\_INS\_2](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_ins_2.vhd). Ty jsou v principu převodníky 6bitové vstupní informace na definovanou posloupnost frekvencí a hlasitostí. Mezi frekvencemi a hlasitostmi přepínají dva čtyřkanálové multiplexery. Tyto údaje pak řídí generátor, který vytváří konkrétní časové vzorky. Použitý generátor generuje trojúhelníkový průběh a je použit shodný generátor, jako je obsažen v bloku [SOUND\_GENERATOR](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/sound_generator.vhd).

Bloky s informacemi o nahrávce dostávají periodicky informaci o stavu čítače CNT\_POINT, což je 6bitový [SIMPLE\_COUNTER](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/simple_counter.vhd), a na základě tohoto stavu mění svou výstupní informaci. Čítač inkrementuje svůj stav co 0,2 s. Aby dostal čítací pulz co 0,2 s, musí být k připojen k hodinovému signálu přes blok [CLOCK\_ENABLE](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/clock_enable.vhd) s nastavenou periodou 20 000 000 hodinových cyklů.

Použitá koncepce byla zvolena, neboť je jednak jednoduchá na realizaci a jednak umožňuje snadné rozšiřování. V současném provedení zvládne jednohlasou melodii, jednu barvu zvuku, 12,6 s času na záznam, tónový rozsah jednu decimu a rozlišení 0,2 s. Nicméně stávající provedení lze snadno rozšiřovat pro delší zvukové záznamy, pro více zvuku i tónový rozsah.

**2.1. Bloky obsahující melodie**

Blok [DEMO\_INS](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_ins.vhd) obsahuje DEMOA, blok [DEMO\_INS\_1](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_ins_1.vhd) obsahuje DEMOB a blok [DEMO\_INS\_2](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_ins_2.vhd) DEMOC.

**Tab. 4 Převodní tabulka bloku** [**DEMO\_INS**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_ins.vhd)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **step** | **freq** | **vol** |  | **step** | **freq** | **vol** |  | **step** | **freq** | **vol** |  | **step** | **freq** | **vol** |
| "000000" | x"C" | x"0" |  | "010000" | x"7" | x"9" |  | "100000" | x"C" | x"7" |  | "110000" | x"0" | x"7" |
| "000001" | x"C" | x"7" |  | "010001" | x"7" | x"6" |  | "100001" | x"B" | x"8" |  | "110001" | x"8" | x"7" |
| "000010" | x"B" | x"8" |  | "010010" | x"7" | x"3" |  | "100010" | x"C" | x"9" |  | "110010" | x"8" | x"6" |
| "000011" | x"C" | x"9" |  | "010011" | x"0" | x"D" |  | "100011" | x"7" | x"9" |  | "110011" | x"5" | x"6" |
| "000100" | x"B" | x"A" |  | "010100" | x"7" | x"D" |  | "100100" | x"A" | x"9" |  | "110100" | x"5" | x"5" |
| "000101" | x"C" | x"B" |  | "010101" | x"8" | x"D" |  | "100101" | x"8" | x"9" |  | "110101" | x"5" | x"0" |
| "000110" | x"7" | x"B" |  | "010110" | x"8" | x"B" |  | "100110" | x"5" | x"9" |  | "110110" | x"7" | x"7" |
| "000111" | x"A" | x"B" |  | "010111" | x"8" | x"9" |  | "100111" | x"5" | x"6" |  | "110111" | x"7" | x"6" |
| "001000" | x"8" | x"B" |  | "011000" | x"8" | x"7" |  | "101000" | x"5" | x"3" |  | "111000" | x"7" | x"5" |
| "001001" | x"5" | x"B" |  | "011001" | x"8" | x"5" |  | "101001" | x"5" | x"0" |  | "111001" | x"7" | x"4" |
| "001010" | x"5" | x"7" |  | "011010" | x"8" | x"3" |  | "101010" | x"0" | x"8" |  | "111010" | x"7" | x"3" |
| "001011" | x"5" | x"5" |  | "011011" | x"8" | x"1" |  | "101011" | x"5" | x"8" |  | "111011" | x"7" | x"2" |
| "001100" | x"5" | x"2" |  | "011100" | x"8" | x"0" |  | "101100" | x"7" | x"8" |  | "111100" | x"7" | x"1" |
| "001101" | x"0" | x"C" |  | "011101" | x"0" | x"4" |  | "101101" | x"7" | x"4" |  | "111101" | x"7" | x"0" |
| "001110" | x"5" | x"C" |  | "011110" | x"C" | x"5" |  | "101110" | x"7" | x"0" |  | "111110" | x"7" | x"0" |
| "001111" | x"7" | x"C" |  | "011111" | x"B" | x"6" |  | "101111" | x"0" | x"8" |  | "111111" | x"7" | x"0" |

**Tab. 5 Převodní tabulka bloku** [**DEMO\_INS\_1**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_ins_1.vhd)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **step** | **freq** | **vol** |  | **step** | **freq** | **vol** |  | **step** | **freq** | **vol** |  | **step** | **freq** | **vol** |
| "000000" | x"C" | x"0" |  | "010000" | x"7" | x"B" |  | "100000" | x"8" | x"8" |  | "110000" | x"A" | x"6" |
| "000001" | x"C" | x"B" |  | "010001" | x"7" | x"6" |  | "100001" | x"C" | x"C" |  | "110001" | x"A" | x"3" |
| "000010" | x"C" | x"6" |  | "010010" | x"5" | x"C" |  | "100010" | x"A" | x"B" |  | "110010" | x"A" | x"0" |
| "000011" | x"A" | x"C" |  | "010011" | x"7" | x"B" |  | "100011" | x"A" | x"9" |  | "110011" | x"A" | x"0" |
| "000100" | x"8" | x"B" |  | "010100" | x"7" | x"6" |  | "100100" | x"9" | x"7" |  | "110100" | x"C" | x"B" |
| "000101" | x"8" | x"6" |  | "010101" | x"8" | x"C" |  | "100101" | x"A" | x"B" |  | "110101" | x"C" | x"6" |
| "000110" | x"A" | x"C" |  | "010110" | x"3" | x"B" |  | "100110" | x"A" | x"9" |  | "110110" | x"A" | x"C" |
| "000111" | x"C" | x"B" |  | "010111" | x"3" | x"5" |  | "100111" | x"5" | x"D" |  | "110111" | x"8" | x"B" |
| "001000" | x"C" | x"6" |  | "011000" | x"3" | x"0" |  | "101000" | x"8" | x"C" |  | "111000" | x"8" | x"6" |
| "001001" | x"D" | x"C" |  | "011001" | x"5" | x"B" |  | "101001" | x"8" | x"A" |  | "111001" | x"A" | x"C" |
| "001010" | x"F" | x"D" |  | "011010" | x"3" | x"B" |  | "101010" | x"7" | x"D" |  | "111010" | x"C" | x"B" |
| "001011" | x"F" | x"7" |  | "011011" | x"1" | x"B" |  | "101011" | x"7" | x"C" |  | "111011" | x"C" | x"6" |
| "001100" | x"F" | x"0" |  | "011100" | x"0" | x"9" |  | "101100" | x"7" | x"A" |  | "111100" | x"D" | x"C" |
| "001101" | x"C" | x"D" |  | "011101" | x"0" | x"7" |  | "101101" | x"8" | x"F" |  | "111101" | x"F" | x"D" |
| "001110" | x"A" | x"D" |  | "011110" | x"3" | x"B" |  | "101110" | x"A" | x"C" |  | "111110" | x"F" | x"6" |
| "001111" | x"8" | x"D" |  | "011111" | x"8" | x"A" |  | "101111" | x"A" | x"9" |  | "111111" | x"F" | x"0" |

**Tab. 6 Převodní tabulka bloku** [**DEMO\_INS\_2**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_ins_2.vhd)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **step** | **freq** | **vol** |  | **step** | **freq** | **vol** |  | **step** | **freq** | **vol** |  | **step** | **freq** | **vol** |
| "000000" | x"3" | x"0" |  | "010000" | x"D" | x"F" |  | "100000" | x"B" | x"1" |  | "110000" | x"A" | x"F" |
| "000001" | x"3" | x"9" |  | "010001" | x"F" | x"F" |  | "100001" | x"3" | x"9" |  | "110001" | x"8" | x"F" |
| "000010" | x"4" | x"A" |  | "010010" | x"B" | x"F" |  | "100010" | x"4" | x"A" |  | "110010" | x"B" | x"F" |
| "000011" | x"6" | x"D" |  | "010011" | x"D" | x"F" |  | "100011" | x"6" | x"D" |  | "110011" | x"D" | x"F" |
| "000100" | x"B" | x"B" |  | "010100" | x"F" | x"F" |  | "100100" | x"B" | x"B" |  | "110100" | x"F" | x"F" |
| "000101" | x"B" | x"8" |  | "010101" | x"F" | x"0" |  | "100101" | x"B" | x"8" |  | "110101" | x"F" | x"0" |
| "000110" | x"6" | x"D" |  | "010110" | x"A" | x"F" |  | "100110" | x"6" | x"D" |  | "110110" | x"D" | x"F" |
| "000111" | x"B" | x"B" |  | "010111" | x"D" | x"F" |  | "100111" | x"B" | x"B" |  | "110111" | x"B" | x"F" |
| "001000" | x"B" | x"9" |  | "011000" | x"D" | x"0" |  | "101000" | x"B" | x"9" |  | "111000" | x"8" | x"F" |
| "001001" | x"6" | x"D" |  | "011001" | x"B" | x"F" |  | "101001" | x"6" | x"D" |  | "111001" | x"D" | x"F" |
| "001010" | x"B" | x"B" |  | "011010" | x"B" | x"D" |  | "101010" | x"B" | x"B" |  | "111010" | x"D" | x"C" |
| "001011" | x"B" | x"8" |  | "011011" | x"B" | x"B" |  | "101011" | x"B" | x"8" |  | "111011" | x"D" | x"9" |
| "001100" | x"B" | x"5" |  | "011100" | x"B" | x"9" |  | "101100" | x"B" | x"5" |  | "111100" | x"D" | x"6" |
| "001101" | x"B" | x"2" |  | "011101" | x"B" | x"7" |  | "101101" | x"B" | x"2" |  | "111101" | x"D" | x"3" |
| "001110" | x"B" | x"0" |  | "011110" | x"B" | x"5" |  | "101110" | x"B" | x"0" |  | "111110" | x"D" | x"0" |
| "001111" | x"B" | x"F" |  | "011111" | x"B" | x"3" |  | "101111" | x"B" | x"F" |  | "111111" | x"D" | x"0" |

**2.2.** [**Čtyřkanálové multiplexery**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/mux.vhd)

Na první pohled je možno si všimnout, že seřazení vstupních signálů je poněkud neobvykle. První demo nahrávka je připojena na port 3, druhá na port 0 a třetí na port 1. Je tomu tak, protože selekční bity odvozují svůj stav od hodnoty mode, konkrétně od jeho dvou nižších bitů. Použitý port tedy odpovídá dvěma nižším bitům: DEMOA (011), DEMOB (100), DEMOC (101). Pro chod zařízení je nutné, aby obsahovalo dva multiplexery, jeden pro frekvenci, druhý pro hlasitost. V obou případech je informace 4bitová a tomu odpovídá i počet bitů multiplexerů.

**Tab. 7 Tabulka popisující funkci čtyřkanálových multiplexerů**

|  |  |  |
| --- | --- | --- |
| sel | output | připojeno |
| "00" | in\_0 | DEMO\_INS\_1 |
| "01" | in\_1 | DEMO\_INS\_2 |
| "10" | in\_2 | - |
| "11" | in\_3 | DEMO\_INS |

**2.2. Převodník** [**FREQ\_CONV**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/freq_conv.vhd)

Jak již bylo poznamenáno výše, generátor trojúhelníku potřebuje 16bitovou informaci o kmitočtu (dlužno rozumět o počtu přeskakovaných vzorků). Pro jednodušší sestavování bloků obsahujících nahrávky byly tóny značeny pouze hexadecimálními čísly od 0 po F. Převod na informaci srozumitelnou generátoru zajišťuje převodník [FREQ\_CONV](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/freq_conv.vhd). Podobnou funkci plní převodník [BOOT\_TO\_FREQ](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/boot_to_freq.vhd), nicméně ten nebylo vhodné užít, poněvadž ten má pořadí tónů přizpůsobené použité maticové klávesnici, tedy na přeházené. Pracovat s takto přeházenými tóny by bylo jednak více časově náročnější, jednak neúčelné. Jak už se v průběhu oživování ukázalo jednotlivé řádky [BOOT\_TO\_FREQ](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/boot_to_freq.vhd) bylo nutno přeházet, a protože byl v [DEMO\_MAIN](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/demo_main.vhd) použit odlišný převodník, nebylo nutno měnit též všechny rutiny zajišťující tvorbu melodií.

**Tab. 8 Převodní tabulka** [**FREQ\_CONV**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/freq_conv.vhd)

|  |  |  |
| --- | --- | --- |
| bcd (hex) | freq | odpovídá |
| 0 | x"0898" | A |
| 1 | x"091B" | A# |
| 2 | x"09A5" | B |
| 3 | x"0A38" | C1 |
| 4 | x"0AD4" | C#1 |
| 5 | x"0B78" | D1 |
| 6 | x"0C27" | D#1 |
| 7 | x"0CE0" | E1 |
| 8 | x"0DA4" | F1 |
| 9 | x"0E74" | F#1 |
| A | x"0F50" | G1 |
| B | x"1039" | G#1 |
| C | x"1130" | A1 |
| D | x"1235" | A#1 |
| E | x"134B" | B1 |
| F | x"1470" | C2 |

**3. Blok** [**MATRIX\_CONTROL**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/matrix_control.vhd)

Obsah obrázku text, diagram, řada/pruh, snímek obrazovky

Popis byl vytvořen automaticky

**Obr. 29 Princip zapojení MATRIX\_DISPLAY**

Tento bloku je založen na principu stavového automatu Moorova typu. Na jeho vstup jsou přivedeny sloupce maticové klávesnice (označeny jako s0 až s3), hodinový signál, který řídí celý proces a resetovací tlačítko BTNC. Na jeho výstup jsou následně přivedeny řádky maticové klávesnice (označeny jako r0 až r3) a dále stavy any a but. Any určuje, zda je stisknuto nějaké tlačítko. V případě stlačení libovolné tlačítko na maticové klávesnici se přiřadí k but příslušné číslo tlačítka, který dále putuje do bloku [**CODER\_1\_FROM\_16**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/coder_1_from_16.vhd). Tento blok však není přímo součástí MATRIX CONTROL.

Celý princip činnosti je popsán pomocí [stavového diagramu](Any%20určuje,%20zda%20je%20stisknuto%20nějaké%20tlačítko.%20V%20takovém%20případě%20se%20přiřadí%20k%20signálu%20but%20příslušné%20číslo%20tlačítka,%20který%20dále%20putuje%20do%20bloku%20CODER_1_FROM_16.%20Tento%20blok%20však%20není%20přímo%20součástí%20MATRIX%20CONTROL.). Stavový automat začíná ve stavu IDLE. Následně přechází do stavu r0, kde se kontroluje signál sig\_en\_1ms. Pokud je tento signál aktivní, automat začne kontrolovat sloupce maticové klávesnice s0 až s3. Pokud je některý ze sloupců roven hodnotě 0, přechází do odpovídajícího stavu. Následně se nastaví stavy any a but, které již byli popsány na počátku. Poté přechází do IDLE V opačném případě stavový automat přechází na další řádek a pokračuje ve stejném procesu, dokud se nedostane na řádek r3. Odtud přechází zpátky do stavu r0.

Stavy jednotlivých tlačítek jsou popsány tabulkou:

**Tab. 9 Tabulka stavů tlačítek**

|  |  |  |
| --- | --- | --- |
| button | any | but |
| s0 | 1 | "0000" |
| s1 | 1 | "0001" |
| s2 | 1 | "0010" |
| s3 | 1 | "0011" |
| s4 | 1 | "0100" |
| s5 | 1 | "0101" |
| s6 | 1 | "0110" |
| s7 | 1 | "0111" |
| s8 | 1 | "1000" |
| s9 | 1 | "1001" |
| s10 | 1 | "1010" |
| s11 | 1 | "1011" |
| s12 | 1 | "1100" |
| s13 | 1 | "1101" |
| s14 | 1 | "1110" |
| s15 | 1 | "1111" |

**3.1. Blok** [**CODER\_1\_FROM\_16**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/coder_1_from_16.vhd)

Tento blok není přímo součástí [MATRIX\_CONTROL](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/matrix_control.vhd), nicméně svou činností s ním přímo souvisí. Jeho úkolem je zajistit, aby při stisknu libovolné klávesy byla rozsvícena právě jedna LED dioda, která odpovídá výšce hraného tónu. Jak již bylo poznamenáno, na výstupu [MATRIX\_CONTROL](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/matrix_control.vhd) nejsou číselné hodnoty seřazeny tak, aby nejvyšší číslo odpovídalo nejvyšší frekvenci. Proto je nezbytné, nejen tyto čtyřbitové číselné hodnoty seřadit, ale i převést je do kódu 1 z 16. Dále musí kodér zajistit, aby v případě, že není stisknuto žádné z tlačítek, nesvítila žádná dioda, proto také blok obsahuje vstup any, který je připojen na výstup any [MATRIX\_CONTROL](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/matrix_control.vhd). Blok realizuje funkci popsanou tabulkou:

**Tab. 10 Převodní tabulka rutiny CODER\_1\_FROM\_16**

|  |  |  |
| --- | --- | --- |
| bcd (hex) | any | ncode (hex) |
| X | 0 | x"0000" |
| x"0" | 1 | x"8000" |
| x"1" | 1 | x"2000" |
| x"2" | 1 | x"1000" |
| x"3" | 1 | x"0400" |
| x"4" | 1 | x"0040" |
| x"5" | 1 | x"4000" |
| x"6" | 1 | x"0800" |
| x"7" | 1 | x"0200" |
| x"8" | 1 | x"0004" |
| x"9" | 1 | x"0010" |
| x"A" | 1 | x"0080" |
| x"B" | 1 | x"0100" |
| x"C" | 1 | x"0001" |
| x"D" | 1 | x"0002" |
| x"E" | 1 | x"0020" |
| x"F" | 1 | x"0008" |

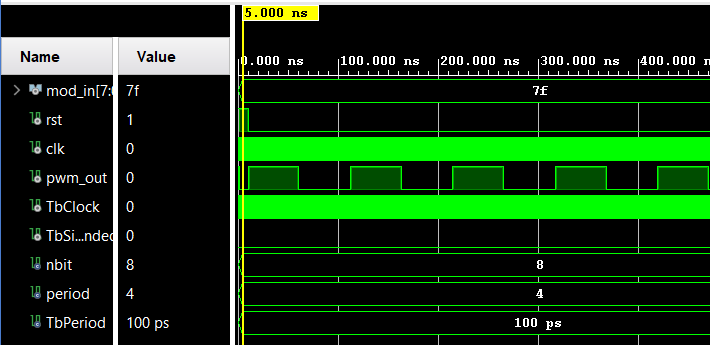
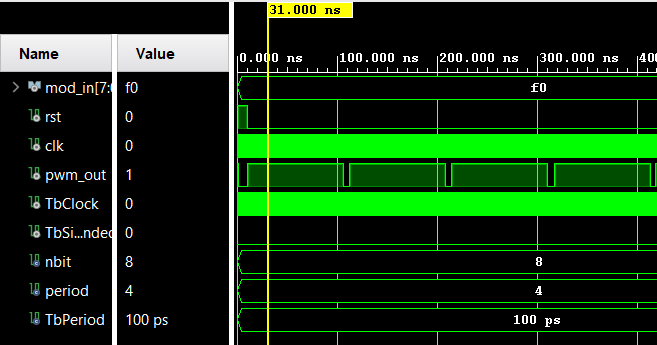
**Simulace komponentů**

Obsah obrázku snímek obrazovky, řada/pruh, Barevnost

Popis byl vytvořen automaticky

**Obr. 30** [**Simulace převodníku prev**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/simprev.vhd)

Jedná se o převodník kódu bcd na 1 z 8 negovaný. Má možnost clear, kdy všechny výstupy budou v 1.



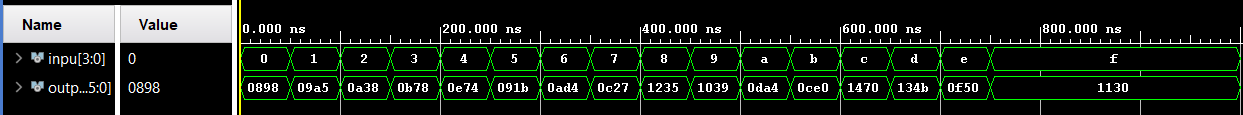
**Obr. 19** [**Simulace PWM\_MOD**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_PWM_mod.vhd) **‒ mod\_in = x“F0”**

**Obr. 20** [**Simulace PWM\_MOD**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_PWM_mod.vhd) **‒ mod\_in = x”7F”**

Obsah obrázku text, snímek obrazovky, displej, software

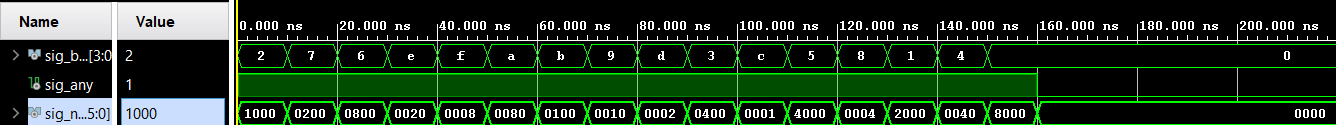
Popis byl vytvořen automatickySimulace bloku [PWM\_MOD](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/PWM_mod.vhd) byla provedena pro více modulačních úrovní, aby byla demonstrována schopnost rutiny generovat PWM modulovaný signál. Pro vyzkoušení další úrovně stačí měnit vstup mod\_in.

**Obr. 21** [**Simulace PWM\_MOD**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_PWM_mod.vhd) **‒ mod\_in = x”0F”**



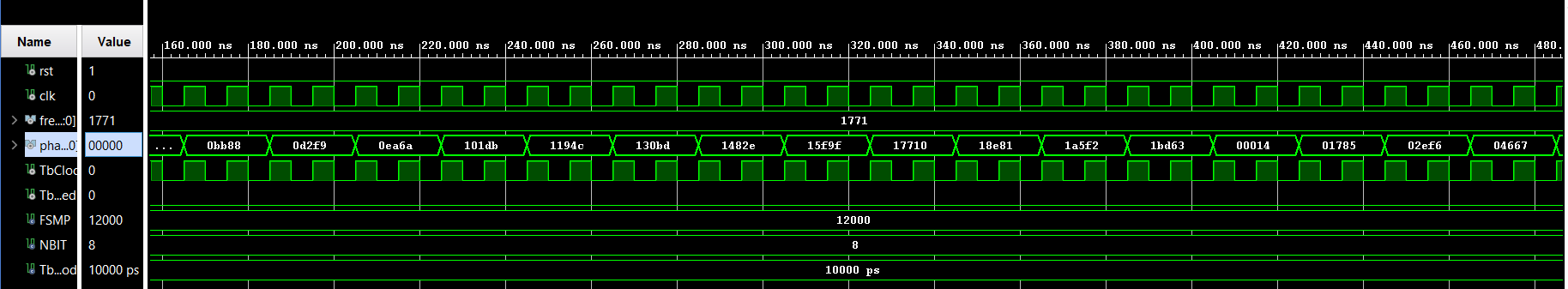
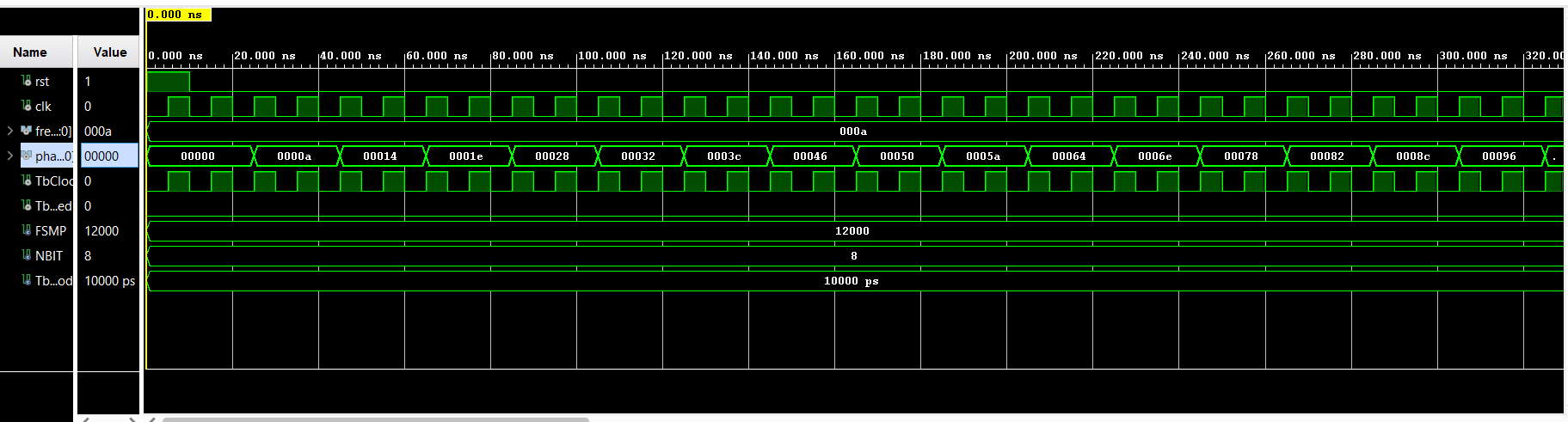
**Obr. 34** [**Simulace BOOT\_TO\_FREQ**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_boot_to_freq.vhd)

Rutina [BOOT\_TO\_FREQ](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/boot_to_freq.vhd) pracuje jako převodník informace z maticové klávesnice na informaci čitelnou pro [SOUND\_GENERATOR](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/sound_generator.vhd).



**Obr. 35** [**Simulace CODER\_1\_FROM\_16**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_coder_1_from_16.vhd)

Jedná se se o kodér z přímé binární hodnoty na přímý kód 1 z 16.



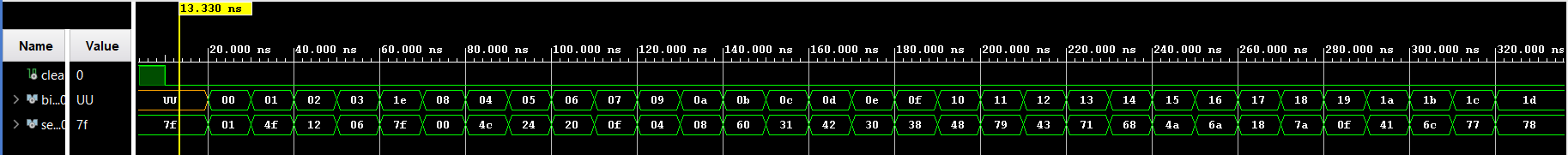
**Obr. 36** [**Simulace COUNTER\_NNUM**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_counter_nnum.vhd) **‒ nahoře freq = “000A”, dole freq = “1771”**

Simulace demonstruje schopnost čítače měnit hodnotu, kterou přičítá, a správné zkrácení cyklu.



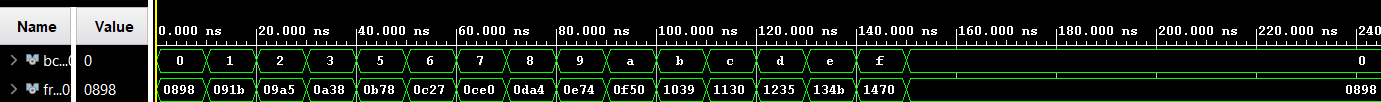
**Obr. 37** [**Simulace BIN2BCD**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_bin2bcd.vhd)

Jedná se o převodník přímého binárního kódu na BCD kód. Zajišťuje, že na displeji se zobrazuje hlasitost jako dekadické, nikoli hexadecimální číslo.



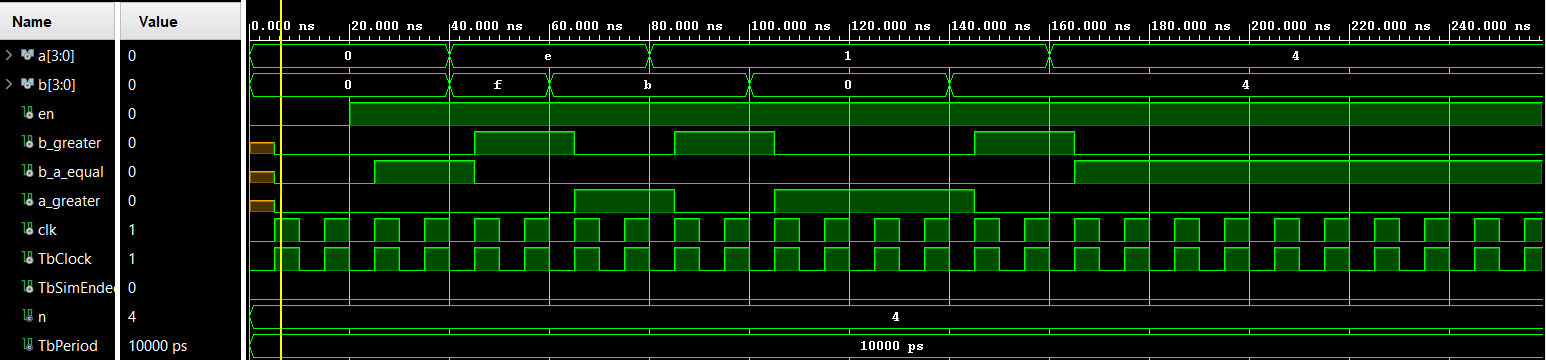
**Obr. 38** [**Simulace BIN2SEG**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_bin2seg.vhd)

Převodník 5bitového kódu na kód segmisegmentového displeje. Umožňuje psát jak číslice, tak písmena.

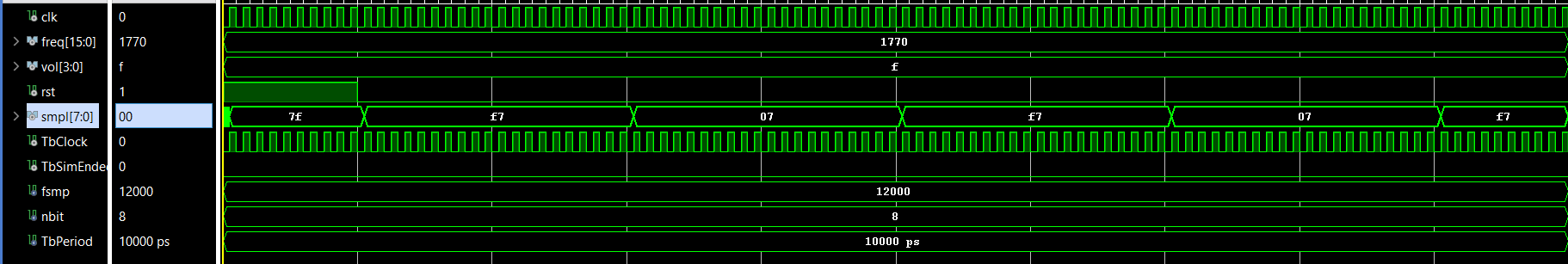


**Obr. 39** [**Simulace FREQ\_CONV**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_freq_conv.vhd)

Převodník 4bitového čísla popisující tón klávesy na 16bitovou informaci pro čítač [COUNTERNNUM](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/counter_nnum.vhd).

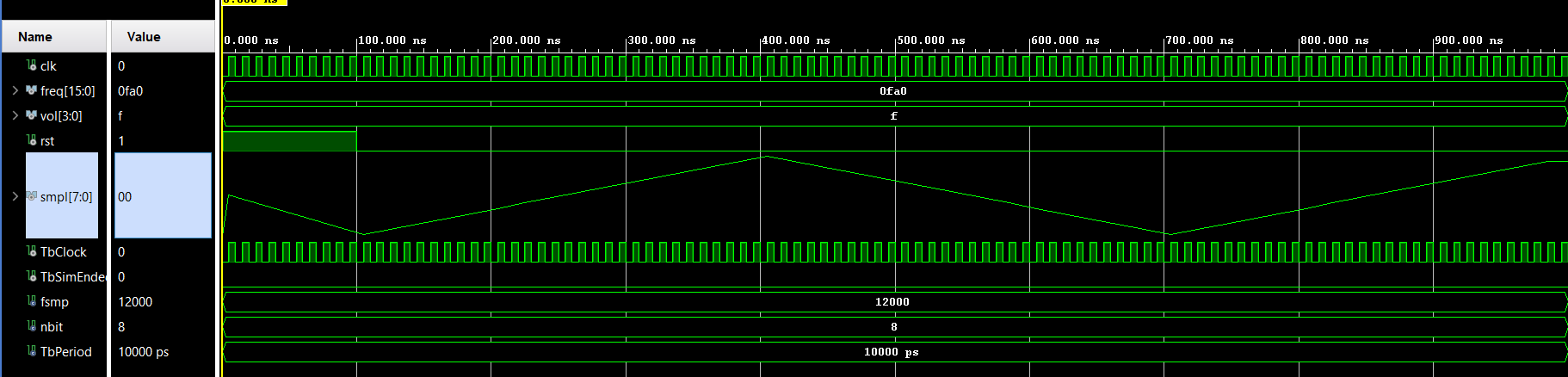


**Obr. 40** [**Simulace KOMPARENBIT**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_kompareNbit.vhd)

Simulace komparátoru, klíčové části PWM modulátoru. Umožňuje užít až tří výstupů, nicméně užívá se pouze jeden.

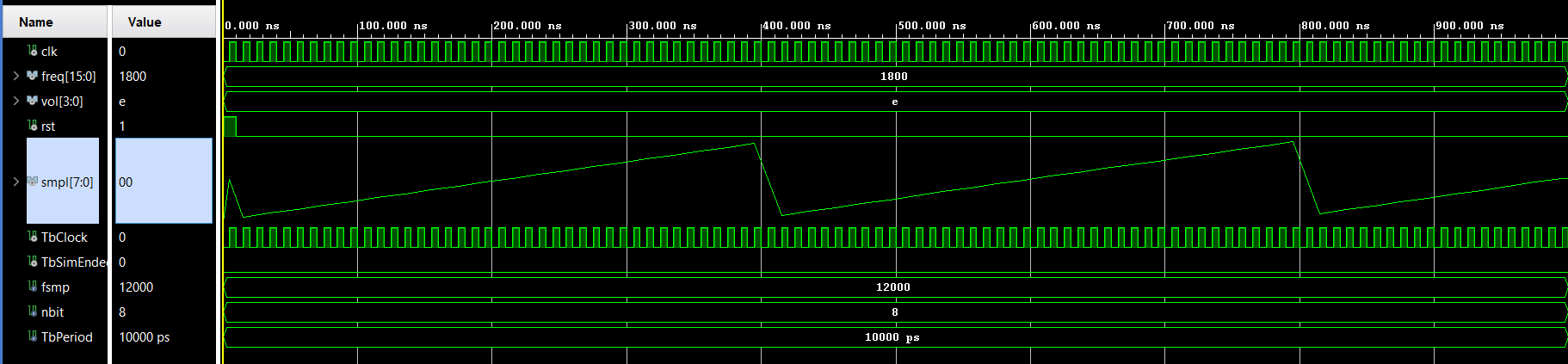
**Obr. 41** [**Simulace REC\_GEN**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_rec_gen.vhd) **‒ freq = x”1771”, vol = x”F”**

Ověření činnosti generátoru obdélníkového průběhu. Není použit analogový výstup, neboť by správně nepopisovat funkci zařízení. Výstupem je smpl.



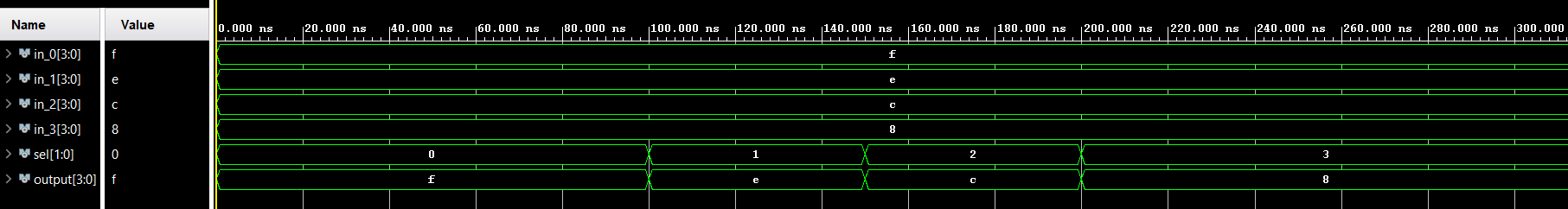
**Obr. 42** [**Simulace TRI\_GEN**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_tri_gen.vhd) **‒ freq = x”1771”, vol = x”F”**

Ověření funkce generátoru trojúhelníkového průběhu. Výstupem je smpl.



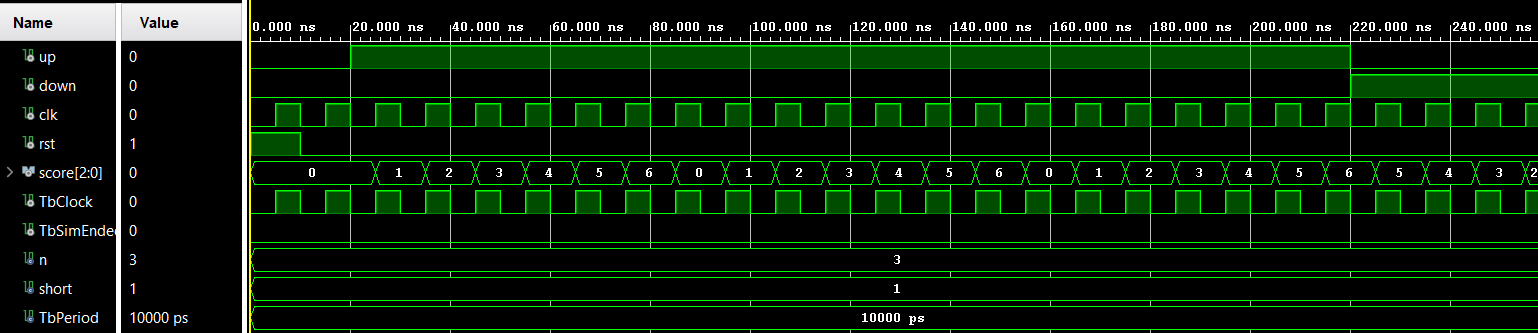
**Obr. 43** [**Simulace SAW\_GEN**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_saw_gen.vhd)**‒ freq = x”1771”, vol = x”F”**

Ověření funkce generátoru pilového průběhu. Výstupem je smpl.



**Obr. 44** [**Simulace MUX**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_muxv.vhd)

Simulace čtyřkanálového multiplexeru. Simulace demonstruje schopnost bloku přepínat mezi jednotlivými vstupy.



**Obr. 45** [**Simulace UPDOWN\_COUNTER**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_updown_counter.vhd)

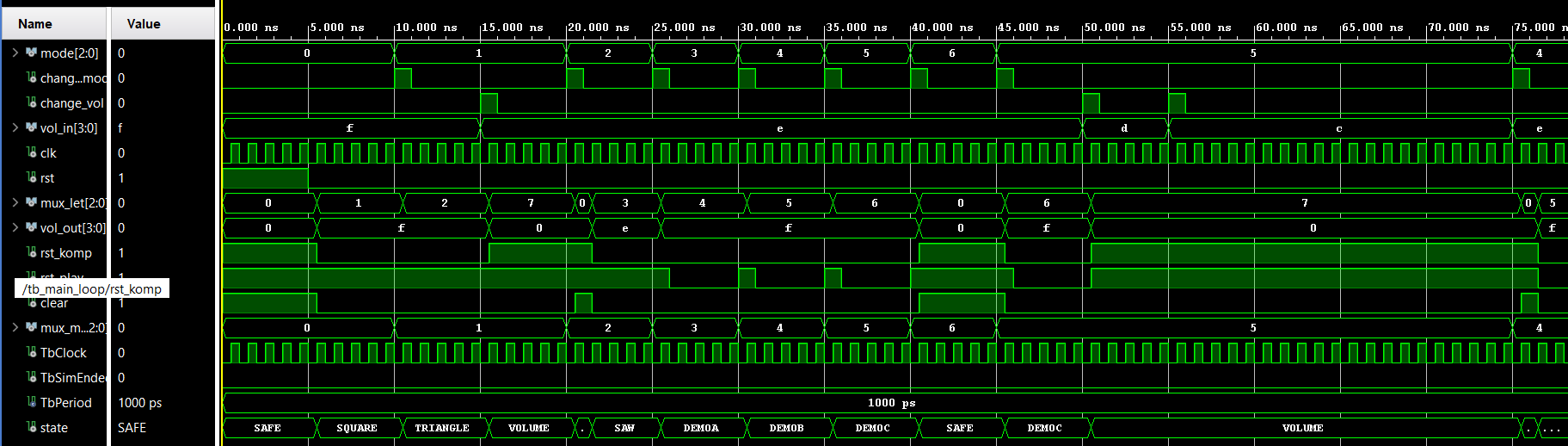
Ověření činnosti vratného čítač s nastavitelným zkrácením čítacího cyklu. Bylo testováno se zkrácení o 1. Nezbytná součást volby módu zařízení.

Obsah obrázku snímek obrazovky

Popis byl vytvořen automaticky

**Obr. 46** [**Simulace UPDOWM\_COUNTER\_LIMITED**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_updown_counter_limited.vhd)

Ověření činnosti vratného čítače, který má zajištěnou ochranu proti přetečení a podtečení. Simulace ukazuje schopnost čítat směrem nahoru i dolů. Taktéž demonstruje neschopnost zařízení přičítat další pulzy po dosažení maximální hodnoty.



**Obr. 47** [**Simulace MAIN\_LOOP**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_main_loop.vhd)

Ověření činnosti hlavního bloku celého zařízení. Simulace demonstruje schopnost tohoto stavového automatu přecházet mezi jednotlivými stavy. Dále ukazuje správně nastavené výstupy. MUX\_LET je oproti vstupu MODE o 1 větší, což odpovídá požadavkům rutiny [DISP\_CON](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/MAIN/Project_PWM.srcs/sources_1/new/disp_control.vhd).

Obsah obrázku snímek obrazovky, Barevnost, řada/pruh

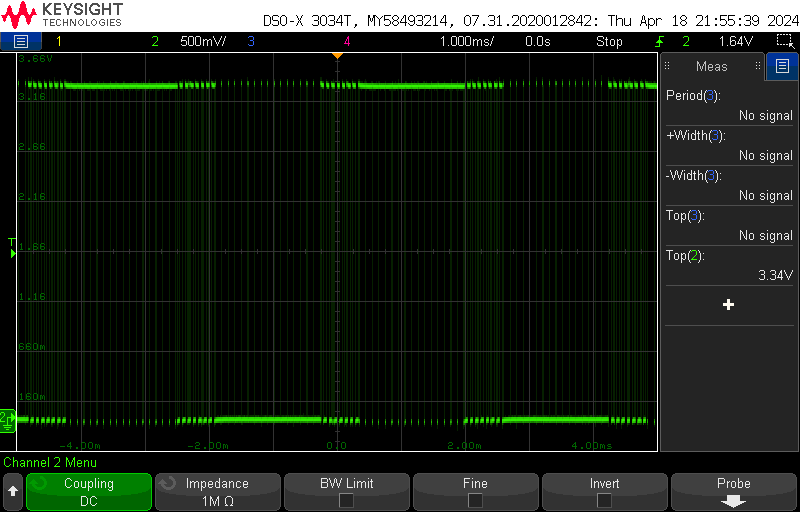
Popis byl vytvořen automaticky

**Obr. 48** [**Simulace MATRIX\_CONTROL**](https://github.com/VojtaKudela/BPC-DE1-topic_4/blob/main/TestBenches/tb_matrix_control.vhd)

Simulace prezentuje činnost rutiny MATRIX\_CONTROL. Rutina nastavuje střídavě nulu na jednotlivé vývody r0 až r1 a čte hodnotu s0 až s3. Na základě hodnoty s0 až s3 mění svůj stav.

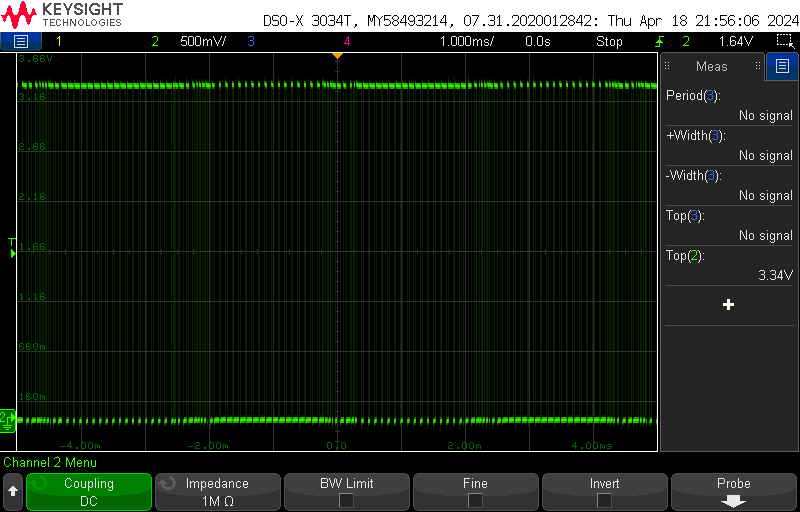
Většina testovacích rutin byla vytvořena automaticky pomocí [1](https://vhdl.lapinoo.net/testbench/).

**Ověření činnosti zařízení**



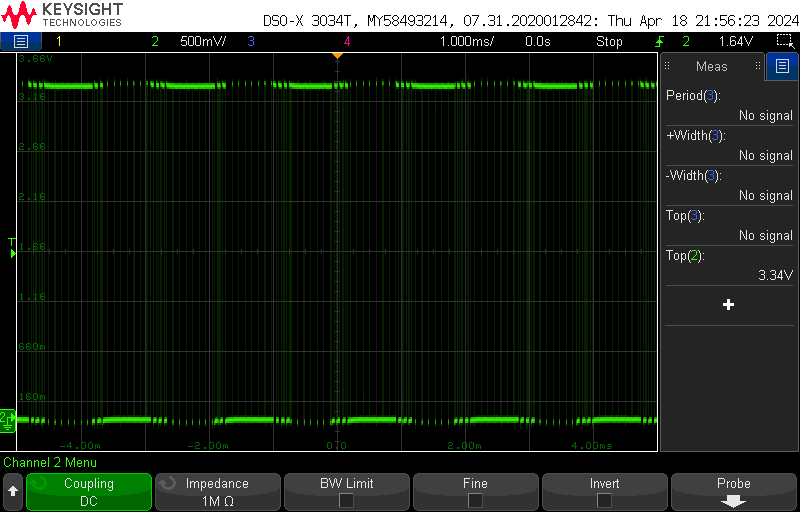
**Obr. 49 Oscilogram ‒ obdélníkový průběh, hlasitost x”F”**

Jedná se o PWM signál modulovaný obdélníkem.



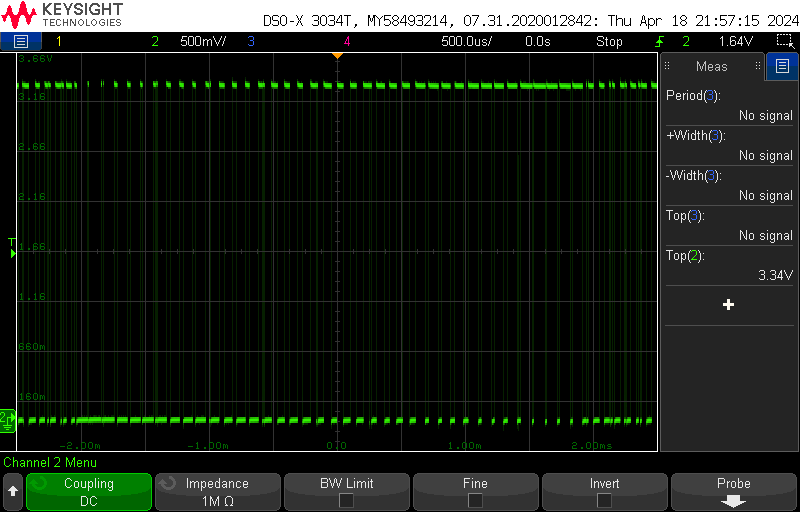
**Obr. 50 Oscilogram ‒ obdélníkový průběh, hlasitost x”8”**

Výstup pro jinou hlasitost. Je zřejmé, že snížení hlasitosti se projevilo menší změnou střídy.



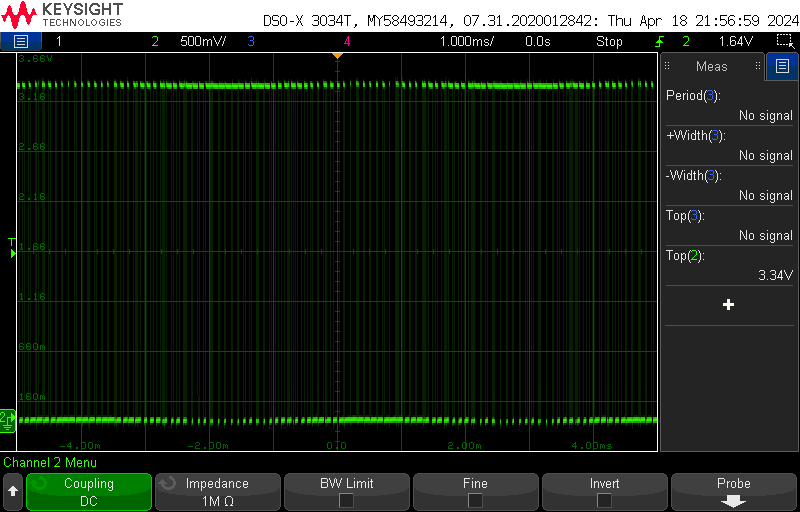
**Obr. 51 Oscilogram ‒ obdélníkový průběh, vyšší kmitočet**

Oscilogram ukazuje, jak se na výstupním průběhu projeví odlišný kmitočet modulačního signálu.



**Obr. 52 Oscilogram ‒ Pilový průběh**

Oscilogram demonstruje schopnost zařízení generovat PWM modulovaný signál modulovaný pilovým průběhem. Že se jedná o pilový průběh je zřejmé, neboť střída modulovaného signálu plynule narůstá od minimální k maximální hodnotě.



**Obr. 53 Oscilogram ‒ Trojúhelníkový průběh**

Oscilogram demonstruje schopnost zařízení generovat PWM modulovaný signál modulovaný trojúhelníkovým průběhem. Že se jedná o trojúhelníkový průběh je zřejmé, neboť střída modulovaného signálu střídavě narůstá a klesá.

**Instrukční list**

Write an instruction manual for your application, including photos and a link to a short app video.

**Reference**

1. [Online VHDL Testbench Template Generator (lapinoo.net)](https://vhdl.lapinoo.net/testbench/)
2. Analogová elektronika 2 – doc. Ing Roman Šotner, PhD.
3. [Pulsně šířková modulace (dhservis.cz)](http://www.dhservis.cz/psm.htm)
4. [František Fuka: Hudba z geekovsko-matematického hlediska - DevFest Praha 2012 (youtube.com)](https://www.youtube.com/watch?v=RQuRFCE5NzI)
5. [Do you hear a people sing?](https://www.youtube.com/watch?v=gYb9sRLUDyM) – Claude-Michel Schönberg
6. [Für Elise](https://www.youtube.com/watch?v=q9bU12gXUyM) – Ludwig van Beethoven
7. [Entertainer](https://www.youtube.com/watch?v=Fxk9qwCFf8s) – Scott Joplin
8. [Microsoft PowerPoint - aud\_prednaska\_1 [režim kompatibility] (vut.cz)](https://moodle.vut.cz/pluginfile.php/827361/mod_resource/content/3/aud_prednaska_1.pdf)
9. [Bc. Václav Štěpán – Funkční generator na principu přímé digitální syntézy](https://dspace.cvut.cz/bitstream/handle/10467/61546/F3-DP-2015-Stepan-Vaclav-diplomka-kompl.pdf?sequence=1)
10. [Nexys A7 - Digilent Reference](https://digilent.com/reference/programmable-logic/nexys-a7/start)
11. [Elektronická učebnice - ELUC (ikap.cz)](https://eluc.ikap.cz/verejne/lekce/924)