Dokumentace

hex – dec konvertor

Jan Kornienko, 2.C

# Úvod

Projekt HEX-DEC konvertor je má závěrečná práce do předmětu HAC pro rok 2019/2020. V této dokumentaci vysvětluji, jak fungují jednotlivé části logického obvodu. Uvádím zde své podklady, podle kterých jsem postupoval při návrhu obvodu.

# Funkce a uživatelské zobrazení

Obsah obrázku snímek obrazovky

Popis byl vytvořen automatickyProgram HEX-DEC konvertor slouží k převodu mezi dvěma číselnými soustavami, a to z šestnáctkové do desítkové. Převodník umí převést až 8bitová čísla. Uživatel zadá pomocí zobrazené klávesnice číslo v hexadecimálním formátu. Zadané číslo se zobrazí na horním displeji a převedené číslo v desítkovém formátu se zobrazí na displeji dolním. Na obrázku 1 můžete vidět uživatelské prostředí programu. V pravém horním rohu se nacházejí dva displeje, určené pro výpis číselných hodnot, každý z nich je označen textem, v jaké soustavě pracuje. Nahoře vpravo je box s názvem Convertor, ve kterém se odehrávají veškeré převody. Tento box má dohromady 17 vstupů z klávesnice a 8 výstupů, které slouží pro výpis dat na oba displeje.

Obrázek 1: Uživatelské rozhraní

# Uživatelský vstup (Klávesnice)

Obsah obrázku hodiny

Popis byl vytvořen automatickyObsah obrázku hodiny

Popis byl vytvořen automatickyObsah obrázku bílá, fotka, černá, hodiny

Popis byl vytvořen automatickyKlávesnice má 17 tlačítek (0-F + DELETE), přes tzv. tunnel se informace přenáší jako vstup do boxu se jménem Convertor (obr. 2 a 3) a následně uvnitř boxu Convertor jako další vstup do druhého boxu se jménem Input (obr. 4).

Obrázek 2: Convertor

Obrázek 3: Input

Obrázek 4: Klávesnice

Obsah obrázku text

Popis byl vytvořen automatickyPo stisknutí tlačítka se přes kontrolovatelný buffer dostane daná hodnota do výstupu a následně se uloží do paměti. Chtěná hodnota je uložena v nástroji s názvem Constant, ten nám umožňuje mít při spuštění čísla jakékoliv hodnoty na místě kde je potřeba, zde např. pro uložení hodnoty tlačítka.

Obrázek 5: Input Box

Výstup Button state slouží pro zaslání informace, zda bylo tlačítko stisknuto, impuls je podmětem pro paměť k zapsání hodnoty. Viditelná hradla na obrázku 5 zajišťují, že když je jedna z hodnot vstupů rovna 1, tak na výstupu bude také 1. Tento způsob přenosu informací jsem zvolil z důvodu předchozích chyb. Používal jsem stejně jako u hodnot tlačítek kontrolovatelný buffer, v této variantě se však vyskytoval error, ten zapříčinilo dostání hodnoty 1 do oblastí ostatních vstupů, kde měla být hodnota 0.

Výstup delete je pouze informace o stisknutí tlačítka pro vymazání obsahu displejů, to je použito následně u smazání paměti.

# Uložení hodnoty

Obsah obrázku hodiny, kreslení

Popis byl vytvořen automatickyV boxu Convertor slouží tři výstupy z Input (Input, Button state, Delete) jako vstupy pro box Register. Na spojení input je před vstupem do boxu paměti jednočíselný 4bitový registr (obr. 6), ten slouží k uchování hodnoty před vstupem do boxu. Tento postup jsem zvolil z důvodu vyskytlé chyby, když tzv. meziregistr chyběl, paměť v boxu nebyla schopna zapsat údaj. Domnívám se, že tomu tak bylo z důvodu krátké doby stisku tlačítka (hodnota Button state 1), který zapříčiňuje zápis do registru. V tomto meziregistru funguje zápis a vymazání stejným způsobem, a to pomocí výstupu Button state a Delete, nenachází se však v boxu, a tak není s uložením problém.

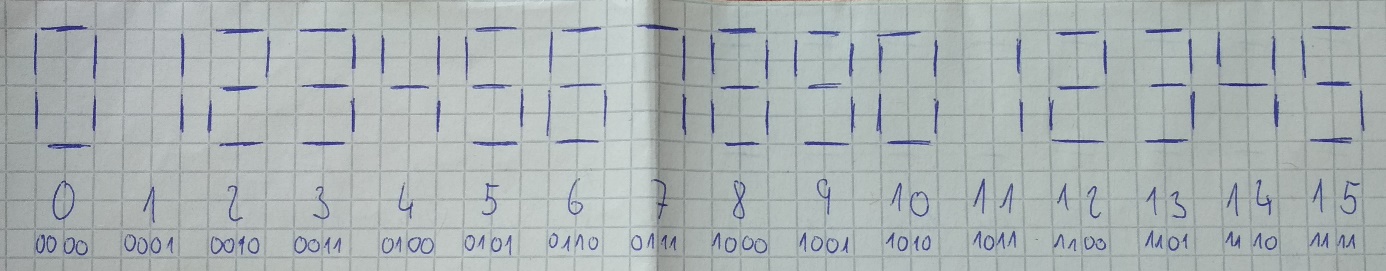
Obrázek 6: Meziregistr a registr

Číselné hodnoty se ukládají do paměti typu register ve stejnojmenném boxu (obr. 7), ten je nastavený pro ukládání dvou 4bitových čísel. Obvod pracuje se třemi vstupy (Input, Button state, Delete button), které slouží ke stejnému účelu jako u zápisu v meziregistru. Rozdíl je v tom, že zde se zapisují dvě čísla, to zapříčiňuje posun v paměti. Jelikož zde nepotřebuji vstupy pro změnu jakékoliv hodnoty (3 vstupy nahoře), použil jsem u nich uzemnění. A naopak stálý přívod hodnoty nula je u povolení zápisu do paměti (vlevo nahoře). Výstup výsledného čísla z paměti (pravý střed) nepoužívám, protože vypisuje pouze druhou hodnotu v paměti. Namísto toho jsem využil okamžitý výpis všech hodnot uložených v paměti (First/Second number). Tyto dvě čísla slouží k výpisu zadaného HEX čísla bez nutnosti převodu. Abych získal výsledné 8bitové číslo pro následující převod, je potřeba dvě předchozí hodnoty spojit. K tomu jsem využil tzv. Splitter, kdy se dva 4bitové splittery spojují s jedním 8bitovým. Získaný údaj jde do výstupu pod názvem Output.

# Obsah obrázku hodiny Popis byl vytvořen automatickyPřevod

Obrázek 7: Register Box

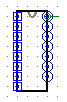
Obsah obrázku snímek obrazovky

Popis byl vytvořen automatickyPro převod číselných soustav jsem zvolil poněkud netradiční způsob. Vytvořil jsem si schéma displeje (obr. 8), ve kterém je vidět, při jakém čísle mají být které části zapnuty. Na základě toho jsem v Logisimu vytvořil pravdivostní tabulku (obr. 9) a funkcí Analyze Circuit jsem vytvořil obvod pro každý displej, každý z nich jsem následně uložil jako boxy (obr. 10).

Obrázek 10: 3 Displeje

Obrázek 9: Pravdivostní tabulka

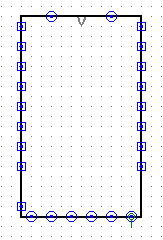
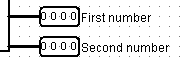
Obrázek 8: Schéma displeje

Obsah obrázku klávesnice

Popis byl vytvořen automatickyVýstupy z nich jsou v přesném pořadí upraveny (obr. 11), tak aby se dal použít splitter k spojení hodnoty pro výstup. Pro DEC zobrazení slouží tedy 6 výstupů pro 3 displeje (obr. 12), jeden výstup pro horní řadu a jeden pro řadu dolní.

Obrázek 11: Upravení výstupů

Obrázek 12: Převod

Pro zobrazení HEX čísla slouží přímo dvě 4bitové hodnoty z výpisu paměti. To jsou další dva výstupy pojmenovány First number a Second number (obr. 13). Aby byl box převodu ve vhodném formátu v uživatelském zobrazení, upravil jsem pozici všech vstupů a výstupů (obr. 14). Samotný box je pak v zobrazení otočen směrem k východu a jsou na něj napojeny všechny potřebné vstupy a výstupy, pomocí pojmenovaných tunelů.

Obrázek 13: Výstupy pro HEX

Obrázek 14: Upravení vstupů a výstupů

# Výpis na displej

## HEX displej

Pro HEX displej jsem zvolil typ výstupu Hex Digit Display z důvodu, že už v šestnáctkové soustavě pracuje v základu. Pro načtení dat jsem tedy zvolil dva vstupy, které jsou označeny First a Second, k rozeznání, pro který panel jsou určena data.

## DEC displej

Obsah obrázku kreslení, hodiny

Popis byl vytvořen automatickyObsah obrázku hodiny

Popis byl vytvořen automatickyObsah obrázku hodiny

Popis byl vytvořen automatickyPro výpis decimálního čísla jsem využil odlišný způsob, a to 7-Segment Display. Učinil jsem tak z důvodu, že samotný převod je založen na principu ovládání tohoto displeje, kdy lze samostatně zapnout jakákoliv část. Vstupy do těchto displejů jsou označeny 1-3 pro určení pořadí displeje a T/B (Top/Bottom), které slouží k orientaci, kdy T značí horní vstupy a B vstupy dolní. Stejně jako u HEX displeje, ani u tohoto jsem nepoužil vstup pro zobrazení desetinné čárky.

Obrázek 16: Convertor

Obrázek 15: HEX displej

Obrázek 17: DEC displej