

Teoretický úvod – ukázková laboratorní úloha

Seznámení s kitem DE10-Lite a jeho obsluhou a programem Intel Quartus

1. Multiplexor 2 na 1

Multiplexor představuje jednoduchý kombinační logický obvod, který obecně multiplexuje n vstupních datových signálů do jednoho společného výstupu na základě stavu řídicího signálu. Nejjednodušším multiplexorem je pak typ 2 na 1, tedy multiplexor se 2 datovými vstupy (obvykle označovanými a , b) a jedním výstupem (obvykle Q) řízený jednobitovým řídicím signálem (obvykle označovaným c). Na základě stavu tohoto dvoustavového řídicího signálu je vždy jeden ze dvou datových vstupů a , b přepínán na společný výstup q . Tab. č. 1 ukazuje pravdivostní tabulku takového multiplexoru.

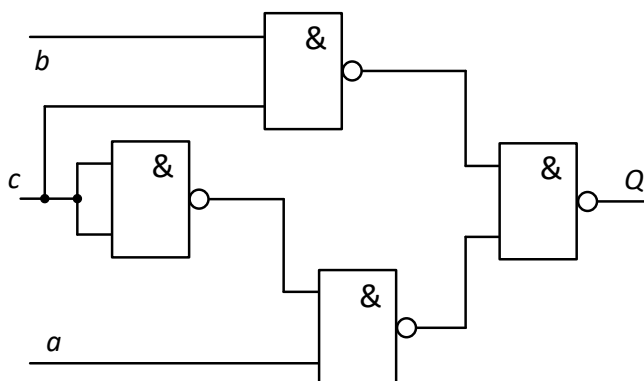
Tab. č. 1: Pravdivostní tabulka základního multiplexoru 2 na 1.

N	c	b	a	Q
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Na základě této pravdivostní tabulky můžeme zapsat výstupní funkci multiplexoru:

$$Q = a\bar{c} \vee bc. \quad (1)$$

A realizovat zapojení pomocí 2vstupých hradel typu NAND dle obr č. 1.



Obr. č. 1: Zapojení základního multiplexoru 2 na 1 pomocí hradel typu NAND.

2. Schematický editor v programu Quartus

Pro obsluhu a programování přípravku DE10-Lite využijeme prostředí (program) Intel Quartus. Ten je přímo určený pro obsluhu FPGA přípravků z produkce Altera-Intel a obsahuje všechny potřebné součásti a ovladače.

Program Quartus lze nainstalovat v placené verzi i variantě, která je zdarma. Neplacená varianta umožňuje jen základní a omezené možnosti práce s přípravky, ale pro potřeby našeho kurzu je zcela dostačující.

Jednou ze součástí programu Quartus je schematický editor. Ten umožňuje navrhovat a realizovat logické obvody v FPGA poli pomocí zakreslení jejich schémat s využitím standardizovaných značek a symbolů jednotlivých logických hradel, pomocných prvků (např. vodičů, uzemnění, napěťových úrovní apod.), ale i celých logických bloků a prvků. Všechny tyto prvky a symboly jsou obsaženy v knihovně, ze které se dají snadno umisťovat přímo na kreslicí plochu a propojovat navzájem jejich vstupy a výstupy pro vytvoření požadovaného zapojení. Editor také obsahuje funkce pro základní kontrolu konzistentnosti nakresleného obvodu a jeho funkce. Následně lze v programu Quartus jednoduše převést nakreslené schéma obvodu na jeho realizaci pomocí jazyka VHDL, které je potom použito pro naprogramování přípravku.¹

3. RTL (dataflow) popis v jazyce VHDL

Jazyk VHDL patří do skupiny tzv. popisných jazyků (description) – realizace obvodů v tomto jazyce je založena na jejich popisu. V jazyce VHDL rozlišujeme 3 způsoby popisu, jak si podrobněji vysvětlíme na přednáškách předmětu a také v navazujících laboratorních úlohách. Tyto tři varianty popisu představují v zásadě 3 různé úrovně abstrakce:

- popis chování (behaviorální popis),
- RTL popis (dataflow popis, popis činnosti obvodu) a
- popis zapojení (strukturální popis).

V této ukázkové úloze využijeme pro realizaci multiplexoru RTL popis, který je vhodný pro realizaci jednoduchých kombinačních obvodů.

¹ V praxi lze schematický editor použít nejen pro kreslení schémat logických obvodů, ale i pro propojování jednotlivých bloků a částí (v jazyce VHDL tzv. component). Nevýhodou schematického editoru je obecně časová náročnost nakreslení zapojení obvodu narůstající s jeho složitostí a také fakt, že je potřeba znát schéma zapojení navrhovaného obvodu. Často je tak realizace obvodu mnohem rychlejší a jednodušší pomocí jeho popisu přímo v jazyce VHDL.

Ten je založen na popisu „toku“, činnosti obvodu, a ukazuje, jak se vstupní logická proměnná průchodem obvodu a kombinacemi s ostatními vstupními proměnnými dostane na daný výstup. V praxi tomu obvykle odpovídá algebraický zápis výstupní funkce obvodu pomocí vstupních proměnných za použití pravidel Booleovy algebry. V jazyce VHDL jsou za tímto účelem definovány základní logické funkce (operátory):

- NOT – negace,
- AND – logický součin,
- OR – logický součet,
- NAND – negovaný logický součin,
- NOR – negovaný logický součet,
- XOR – exkluzivní logický součet,
- XNOR – negovaný exkluzivní logický součet.

Nejvyšší prioritu při vykonávání uvedených operací má negace NOT a všechny zbývající operace mají nižší a stejnou prioritu, tedy i logické násobení a logické sčítání. To odlišuje jazyk VHDL od standardních pravidel algebry. Jazyk VHDL vyhodnocuje zapsané výrazy ve směru zleva doprava a při zápisu logické funkce je potřeba rozlišit priority pomocí kulatých závorek. Pokud se v rámci jednoho výrazu vyskytují různé logické operátory (s výjimkou negace NOT), je nutné, aby tyto operátory byly vždy odděleny pomocí závorek a bylo tedy na první pohled patrné, která operace má prioritu. Více a podrobněji si k popisům v jazyce VHDL, RTL popisu a jeho pravidlům řekneme na přednáškách předmětu a v některých dalších úlohách.

Na závěr tohoto textu se vrátíme k původnímu úkolu, tedy k realizaci multiplexoru pomocí dataflow (RTL) popisu v jazyce VHDL. Jednotlivé části VHDL kódu si podrobněji vysvětlíme na vlastním laboratorním cvičení a v rámci přednášek předmětu.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity multiplexor_vhdl is
port (a,b,c : in std_logic;
      Q : out std_logic);
end multiplexor_vhdl;

architecture RTL of multiplexor_vhdl is
begin
Q<=(a and not c) or (b and c);
end RTL;
```