

Logika układów cyfrowych lab.

Prowadzący: Mgr inż. Antoni Sterna (E02-38m, wtorek 17:05)

sprawozdanie 2 - 2017.10.17

Jakub Dorda 235013
Marcin Kotas 235098

22 października 2017

L^AT_EX

1 Wprowadzenie/cel ćwiczeń

Celem pierwszego zadania było stworzenie transkodera naturalnego kodu binarnego 3 bitowego wykonującego dla każdego przypadku operację dodawania +3. Kolejnym zadaniem było wykonanie subtraktora pełnego 1 bitowego. Trudność w drugim zadaniu polegała na ograniczeniu implementacji tylko do dwuwejściowych bramek NAND. Obydwa układy zostały sprawdzone w symulatorze oraz przez wykonanie ich na zestawie prototypowym w czasie zajęć w laboratorium.

2 Transkoder naturalnego kodu binarnego na kod +3

2.1 Tabela prawdy i tablice Karnaugh:

Tabela 1: Tabela Prawdy

| a | b | c | y ₃ | y ₂ | y ₁ | y ₀ |
|---|---|---|----------------|----------------|----------------|----------------|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

Tabela 2: Tablica Karnaugh dla y₃

| c \ ab | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |

Tabela 3: Tablica Karnaugh dla y₂

| c \ ab | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 |

Tabela 4: Tablica Karnaugh dla y₁

| c \ ab | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |

Tabela 5: Tablica Karnaugh dla y₀

| c \ ab | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |

2.2 Minimalizacje:

$$y_3 = ab + ac = \overline{\overline{ab} \cdot \overline{ac}}$$

$$y_2 = a\bar{b}\bar{c} + \bar{a}b + \bar{a}c = \overline{\overline{a\bar{b}\bar{c}} \cdot \overline{\bar{a}b} \cdot \overline{\bar{a}c}}$$

$$y_1 = bc + \bar{b}\bar{c} = \overline{\overline{bc} \cdot \overline{\bar{b}\bar{c}}}$$

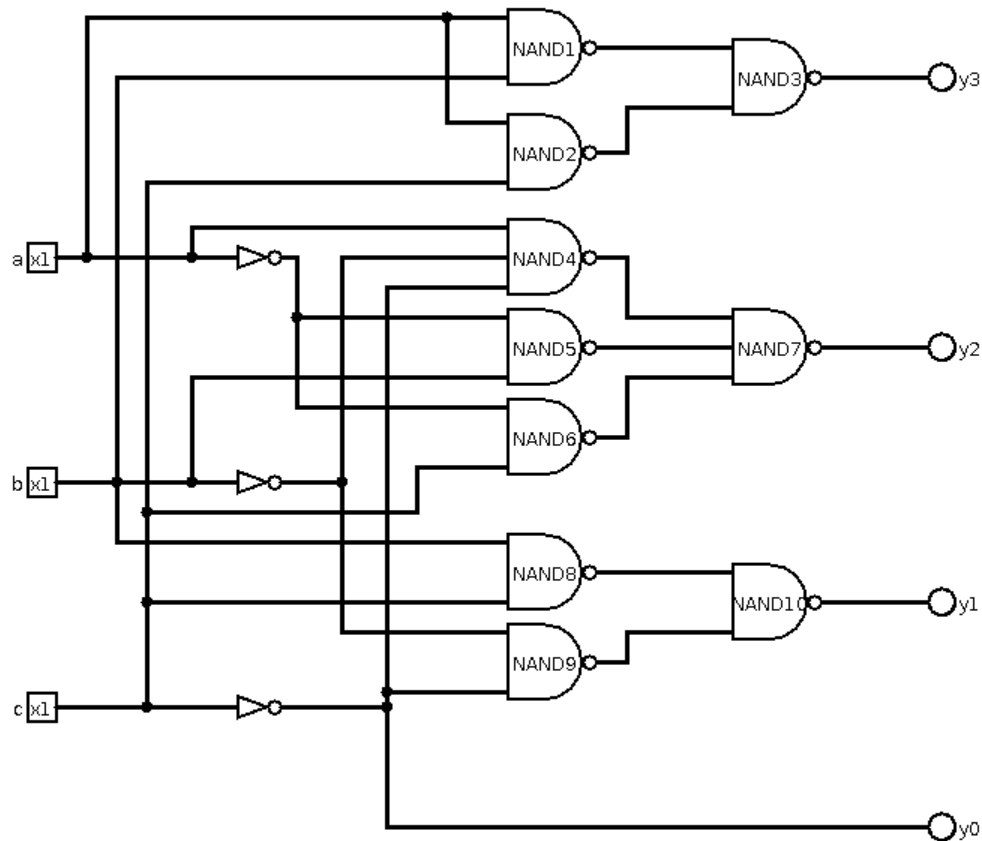
$$y_0 = \bar{c}$$

2.3 Użyte wzory:

$$\overline{a \cdot b} = \bar{a} + \bar{b} \quad (1)$$

$$\overline{a + b} = \bar{a} \cdot \bar{b} \quad (2)$$

2.4 Schemat układu:



Schemat 1. Układ transkodera na bramkach NAND

3 Subtraktor pełny 1 bitowy (na NAND 2 wejściowych)

3.1 Tabela prawdy i tablice Karnaugh:

A_i - odjemna, B_i - odjemnik, C_i - pożyczka od poprzedniego bitu

D_i - bit różnicy/wynik, C_{i+1} - bit pożyczki/przeniesienie

3.2 Minimalizacje:

$$D_i = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + ab\bar{c} = \bar{a}(\bar{b}c + b\bar{c}) + a(\bar{b}\bar{c}) = \overline{\overline{\bar{a}(\bar{b}c + b\bar{c}) \cdot a(\bar{b}\bar{c})}} = \overline{\bar{a}(\bar{b}c \cdot b\bar{c}) \cdot a(\bar{b}\bar{c})} = \overline{\bar{a}(\bar{b}c \cdot \bar{b}\bar{c}) \cdot a(\bar{b}\bar{c})}$$

$$C_{i+1} = \bar{a}b + \bar{a}c + bc = \bar{a}b + c(\bar{a} + b) = \overline{\overline{\bar{a}b \cdot c(\bar{a} + b)}} = \overline{\bar{a}b \cdot c(\bar{a} \cdot \bar{b})} = \overline{\bar{a}b \cdot c(a \cdot b)}$$

3.3 Użyte wzory:

$$\overline{a \cdot b} = \bar{a} + \bar{b} \quad (3)$$

$$\overline{a + b} = \bar{a} \cdot \bar{b} \quad (4)$$

Tabela 6: Tabela Prawdy

| A | B | C | D_i | C_{i+1} |
|---|---|---|-------|-----------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

$$D_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = A_i < (B_i + C_i)$$

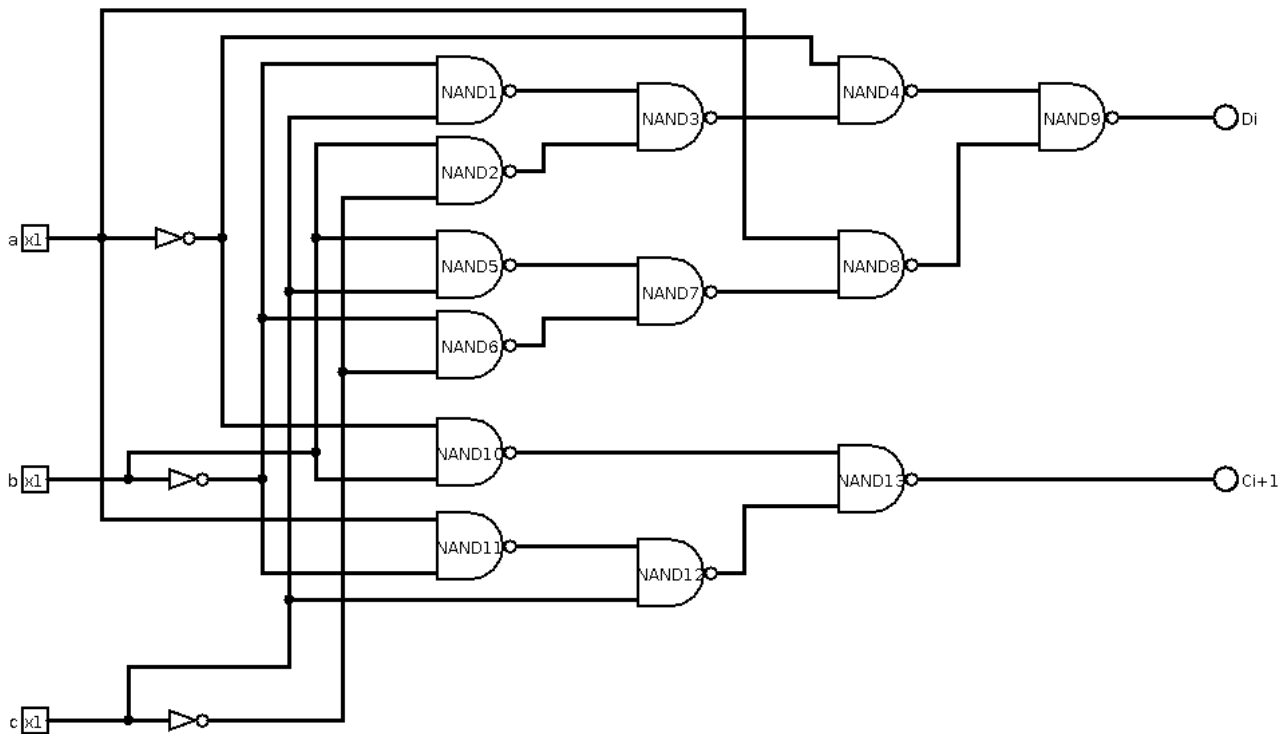
Tabela 7: Tablica Karnaugh dla D_i

| $C \backslash AB$ | 00 | 01 | 11 | 10 |
|-------------------|----|----|----|----|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

Tabela 8: Tablica Karnaugh dla C_{i+1}

| $C \backslash AB$ | 00 | 01 | 11 | 10 |
|-------------------|----|----|----|----|
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

3.4 Schemat układu:



Schemat 1. Układ subtraktora na dwuwejściowych bramkach NAND

4 Wnioski/podsumowanie

W celu sprawdzenia poprawności działania należało przeprowadzić testy dla wszystkich możliwych kombinacji wejść w tym przypadku dla obu układów było to $2^3 = 8$. Obydwa układy działały poprawnie, aczkolwiek w przypadku pierwszego zadania możliwa była dalsza minimalizacja oraz użycie innych bramek w celu uproszczenia finalnego układu.