**Звіт**

**Практична робота “Побудова ALU”**

**Варіант 20**

Кальченко Данило

* Файл *alu.circ* містить готову схему для побудови ALU, що виконує 2-бітне віднімання (у доповнювальному коді) двох чисел А та В, а також виконує операцію порівняння A < B.
* Файл *схема.png* – принципова схема підключення.
* (Побудова виконувалась в симуляторі Logisim-evolution)
* На даній схемі входи А0, А1 (або В0, В1) містять значення відповідних бітів чисел А та В (від молодшого до старшого). Inc – можливе позичання біту.
* Inc – завжди дорівнює (+1 для доповнювального коду)
* Вхід Ctrl виконує переключення між операціями:
* 0 віднімання
* 1 порівняння
* Виходи S0, S1 відповідають значенню розрядів різниці двох чисел (від молодшого до старшого). Cout – можливе позичання.
* Вихід Lt:
* 1 якщо A < B
* 0 якщо A >= B

**Звіт**

**Практична робота “Побудова ALU”**

**Варіант 20**

Кальченко Данило

* Файл *alu.circ* містить готову схему для побудови ALU, що виконує 2-бітне віднімання (у доповнювальному коді) двох чисел А та В, а також виконує операцію порівняння A < B.
* Файл *схема.png* – принципова схема підключення.
* (Побудова виконувалась в симуляторі Logisim-evolution)
* На даній схемі входи А0, А1 (або В0, В1) містять значення відповідних бітів чисел А та В (від молодшого до старшого). Inc – можливе позичання біту.
* Inc – завжди дорівнює (+1 для доповнювального коду)
* Вхід Ctrl виконує переключення між операціями:
* 0 віднімання
* 1 порівняння
* Виходи S0, S1 відповідають значенню розрядів різниці двох чисел (від молодшого до старшого). Cout – можливе позичання.
* Вихід Lt:
* 1 якщо A < B
* 0 якщо A >= B