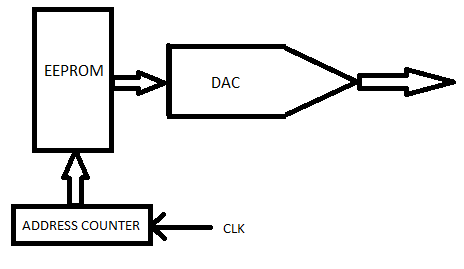
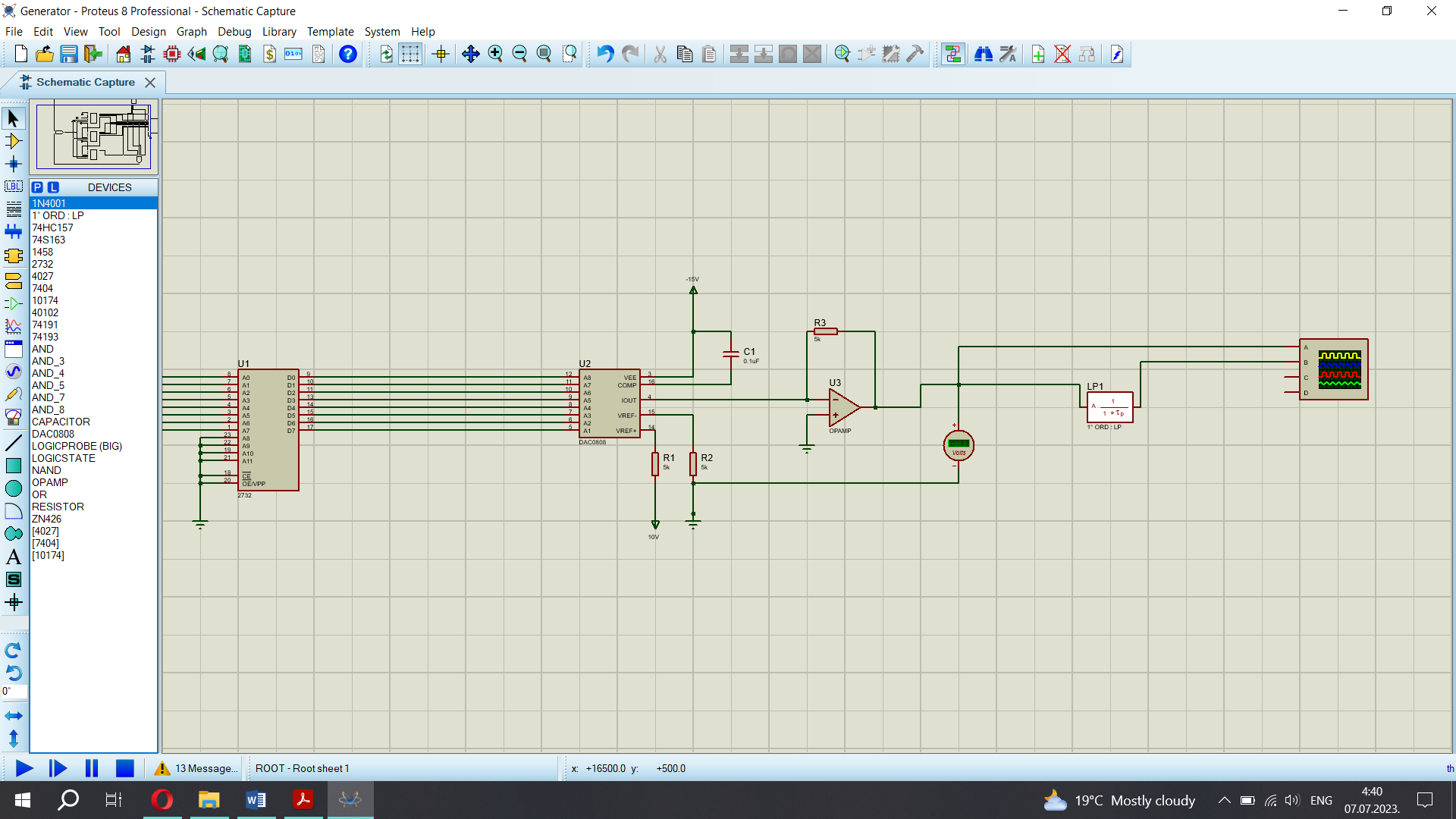
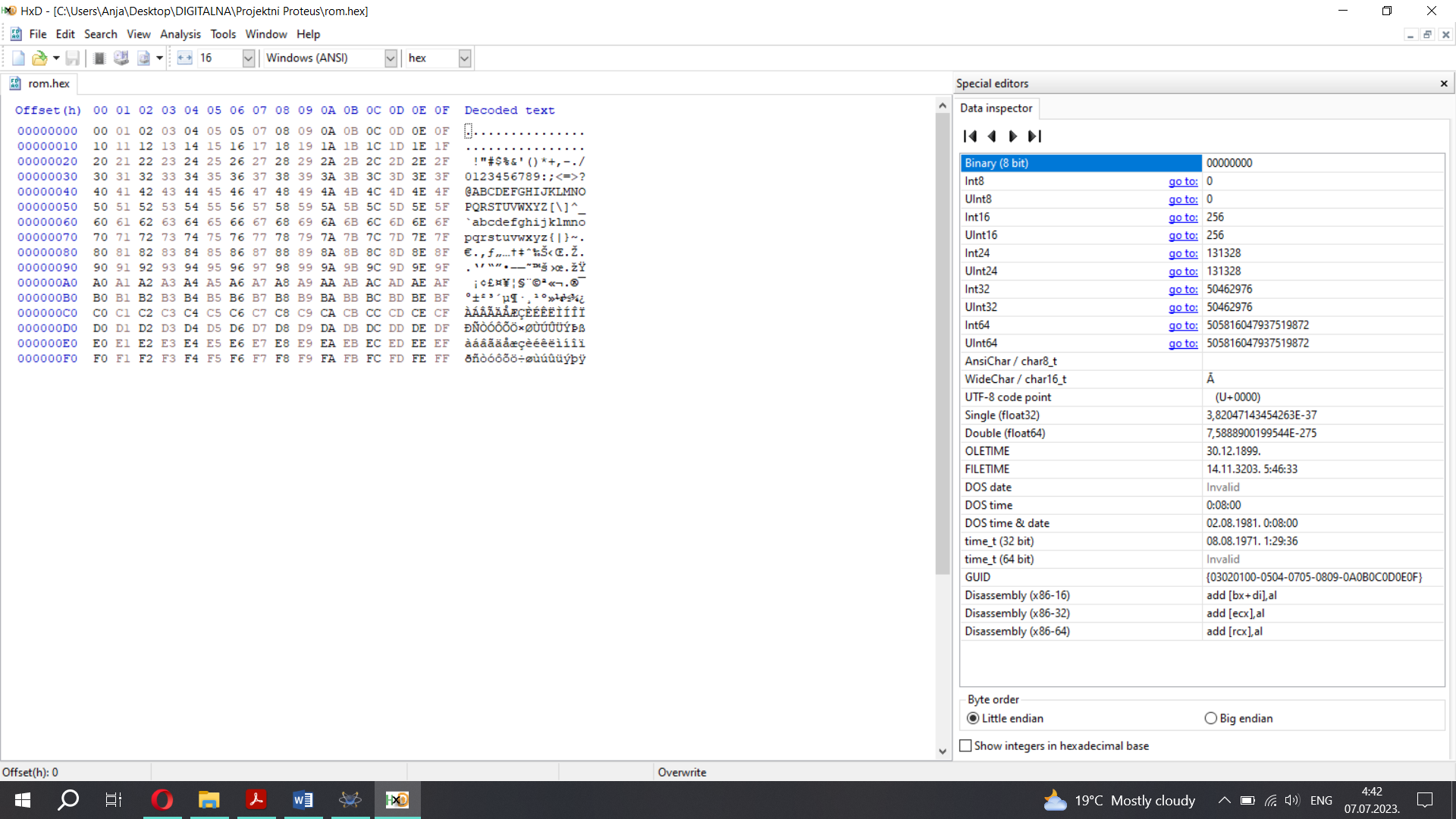
# ГЕНЕРАТОР ФУНКЦИЈА реализован помоћу ROM и DAC компонената





Као EEPROM компонента кориштена је 32К (4Кx8) 2732 чији меморијски садржај одговара садржају са слике rom.hex.

Адресирањем различитих адреса на улазу 2732, на излазу исте компоненте добићемо различите кобинације активних излаза, почевши од адресе

0000 0000 до 1111 1111.

Активација адресне комбинације

0000 0000 0000 ће на излазу резултовати комбинацијом 0000 0000

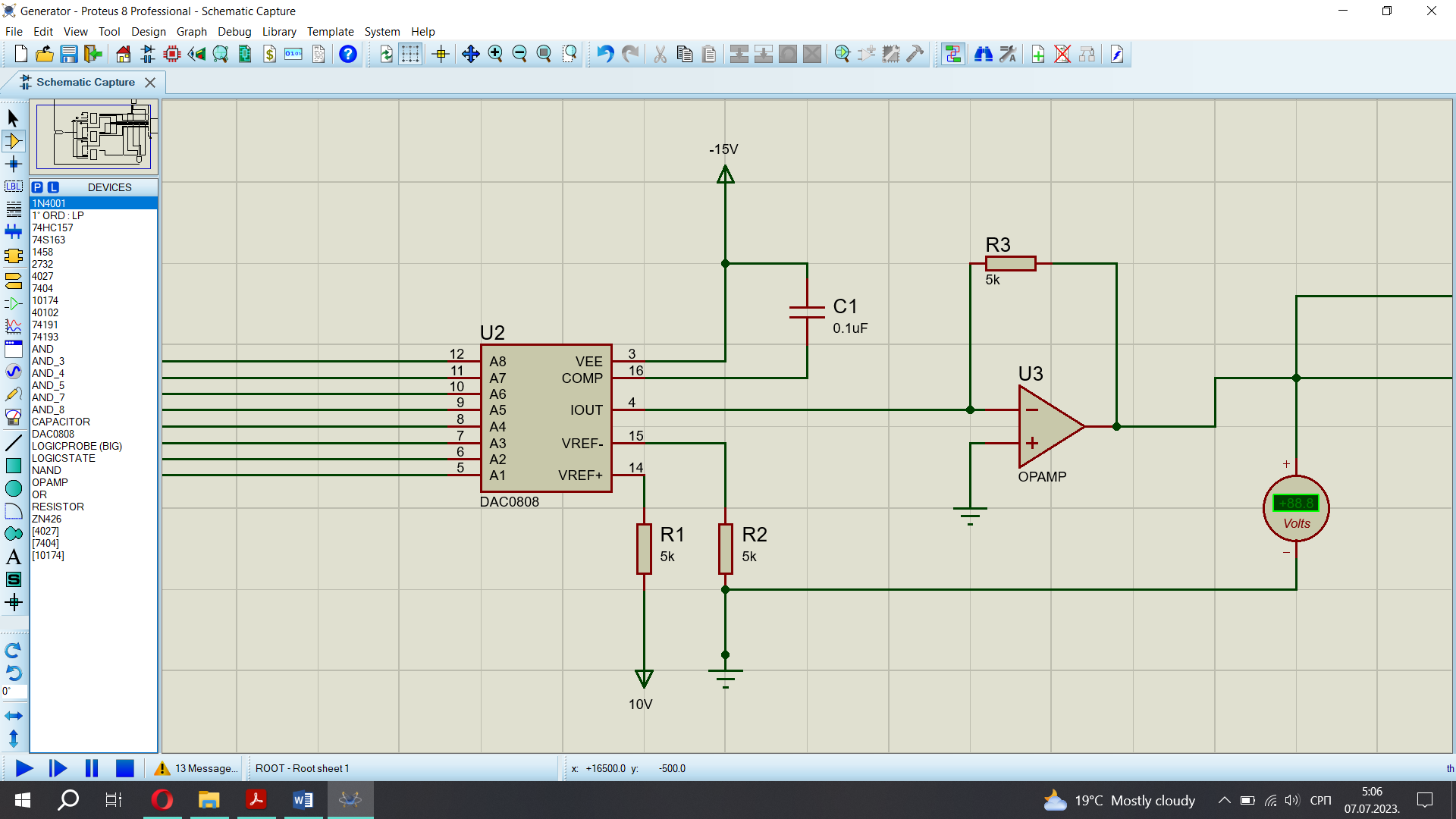
0000 0000 0010 ће на излазу резултовати комбинацијом 0001 0000

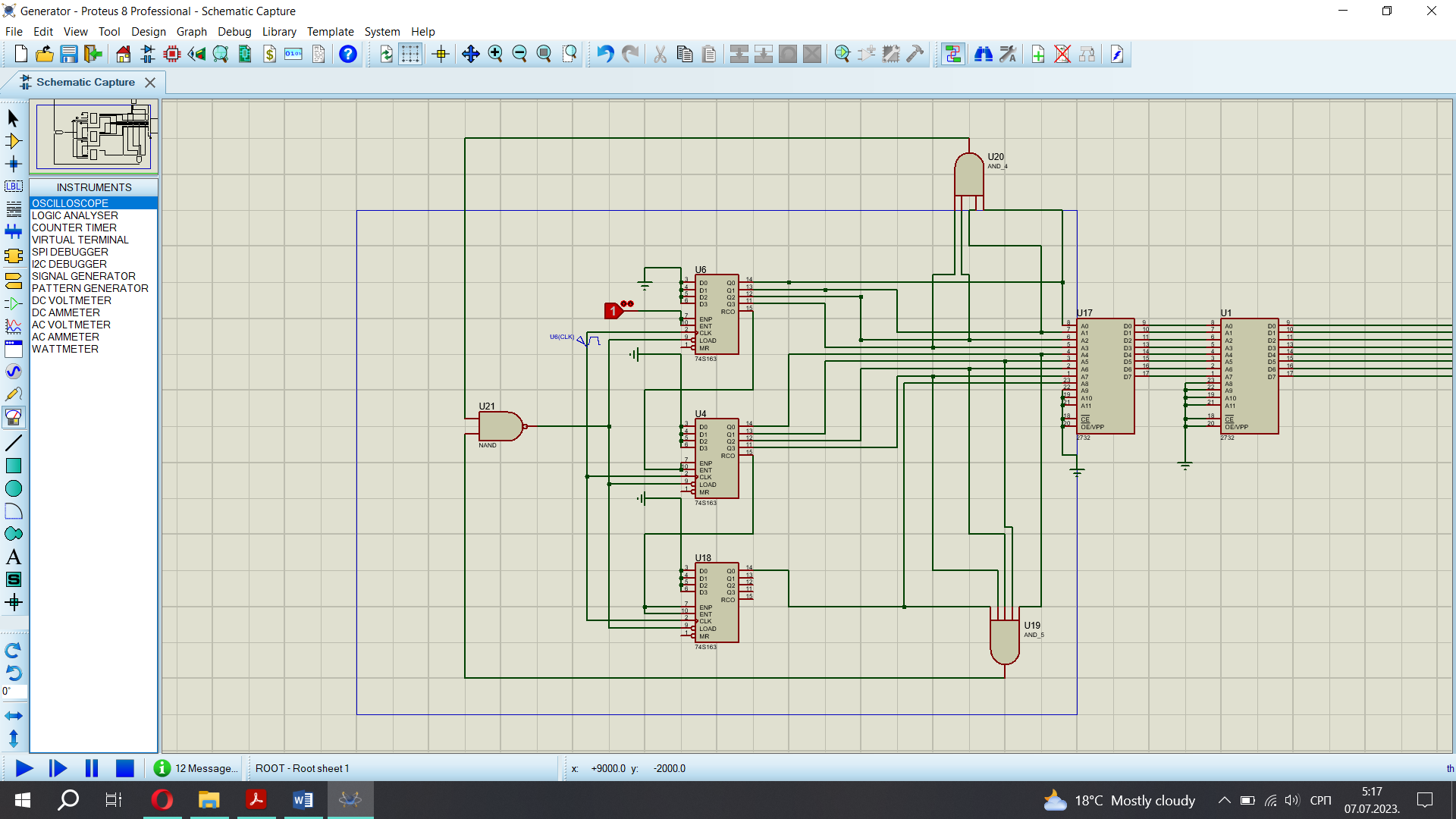
0000 0000 00А4 ће на излазу резултовати комбинацијом 1010 0100

0000 0000 00FF ће на излазу резултовати комбинацијом 1111 1111 ...

Пошто нам је неопходно да адресирамо све бинарне комбинације од 00 до FF , за то ће бити потребно 8 улаза 2732 компоненте (А0 – А7) те преостале пинове (А8 – А11) везујемо за масу. На овај начин адресирамо укупно 256 различитих бинарних комбинација.

Бинарне комбинацје са излаза 2732 компоненте доводимо на улаз DAC0808 који ће за различите бин. комбинације на свом улазу генерисати различите напонске нивое.

Стандардан начин примјене DAC0808 дат је на слици поред.

За адресирање кориштен је дигитални склоп дат на слици лијево. Ради се о класичном бројачу модула М = 511 који броји 0000 0000 0000 0000 –

0000 0001 1111 1111 (00 00-01 FF).

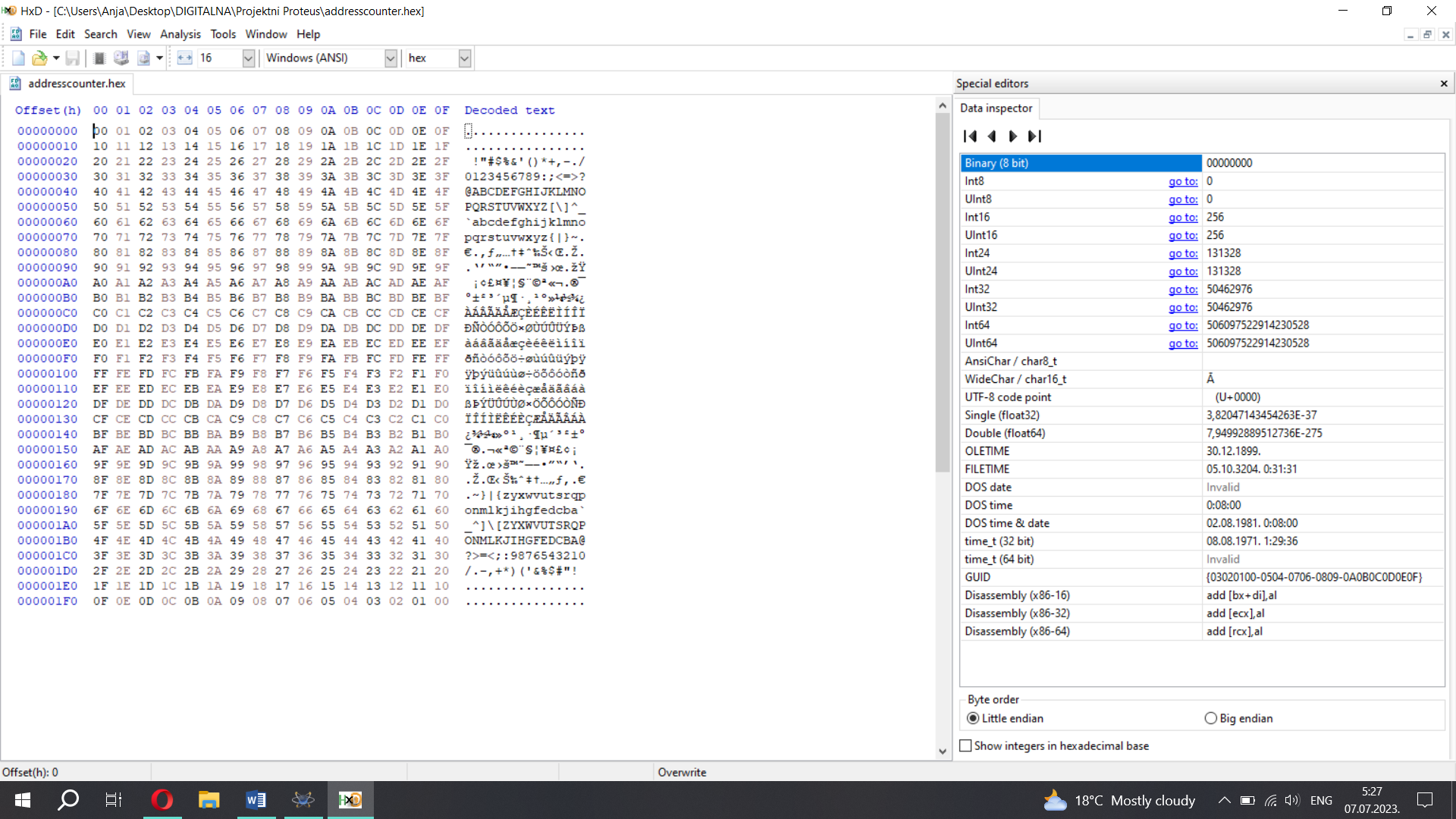
Када се одброји адресна комбинација 01 FF, помоћно логичко коло састављено од “И” кола ће пребацити бројачке компоненте 74S163 у мод паралелног синхроног уписа

(Load = 0 , D3D2D1D0 = 0000) те ће се успоставити почетно стање.

Користимо још једну ROM компоненту чији је адресни садржај дат на слици испод. Пошто је кориштен бројач унапријед а не обострани, управо ова ROM компонента ће нам обезбиједити постепен раст, а затим и постепен пад бинарних комбинација

00 -> … -> D7 -> … -> **FF -> FE -> FD** -> FE -> … -> 42 -> … -> 00

У једном бројачком циклусу ће се проћи 2 пута кроз исту адресну комбинацију, 1. пут током раста, а 2. пут током опадања бин. комбинација.



# Симулација

