ЗМІСТ

[ВСТУП 3](#_Toc156797118)

[РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ 5](#_Toc156797119)

[1.1 Керування семисегментним індикатором. 5](#_Toc156797120)

[1.2 Робота з терміналом середовища “Intel FPGA Monitor Program” 6](#_Toc156797121)

[РОЗДІЛ 2. СТРУКТУРНА СХЕМА ПРИСТРОЮ 7](#_Toc156797122)

[2.1 Структурна схема пристрою та її опис 7](#_Toc156797123)

[РОЗДІЛ 3. СТВОРЕННЯ ПРИСТРОЮ 9](#_Toc156797124)

[3.1. Створення та симуляція модуля «Indicator Driver» 9](#_Toc156797125)

[3.2. Проєкт в середовищі «Quartus» 11](#_Toc156797126)

[3.3. Проєкт в середовищі «Intel FPGA Monitor Program» 15](#_Toc156797127)

[ВИСНОВОК 17](#_Toc156797128)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 18](#_Toc156797129)

# ВСТУП

В межах даної курсової роботи було вирішено розробити систему, що би дозволила користувачу керувати сукупністю семисегментних індикаторів через термінал комп’ютера.

Програмний термінал – це широко розповсюджений інструмент роботи з комп’ютерними чи програмними системами. Ось для приколу вигляд терміналу, що надається операційною системою “Windows” (Рис. 1) чи програмного забезпечення “Intel FPGA Monitor Program” (Рис. 2).

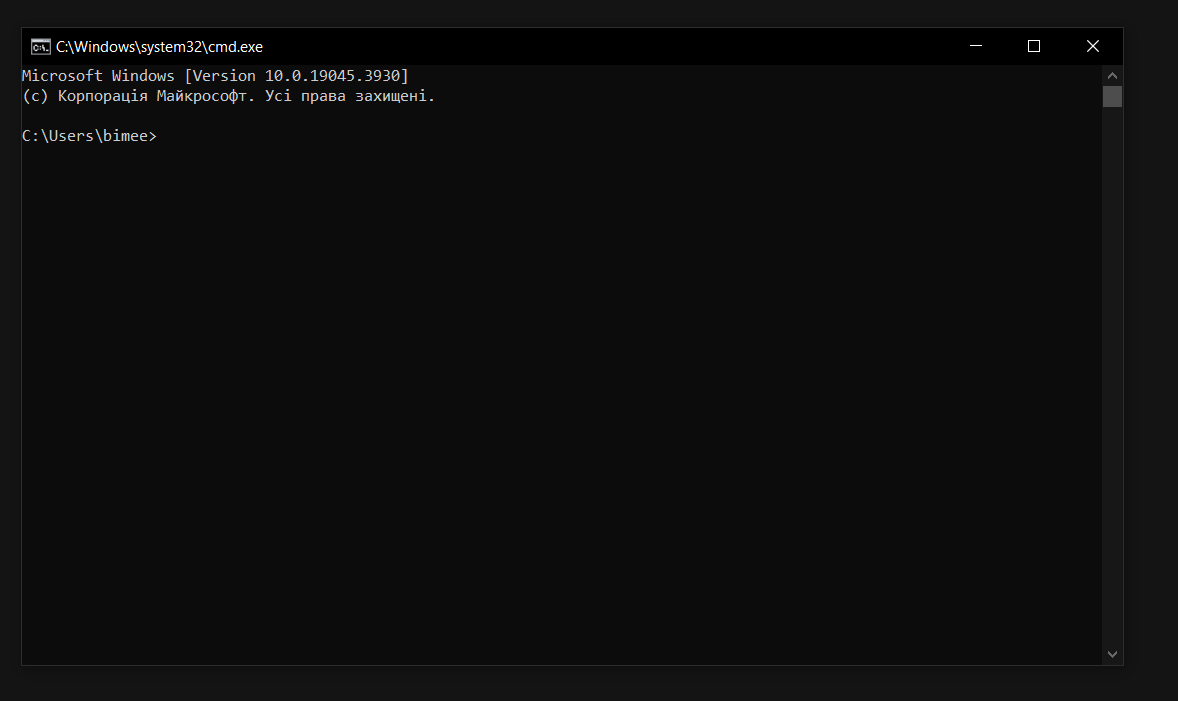


Рис. 1. Програмний термінал в операційній системі “Windows».

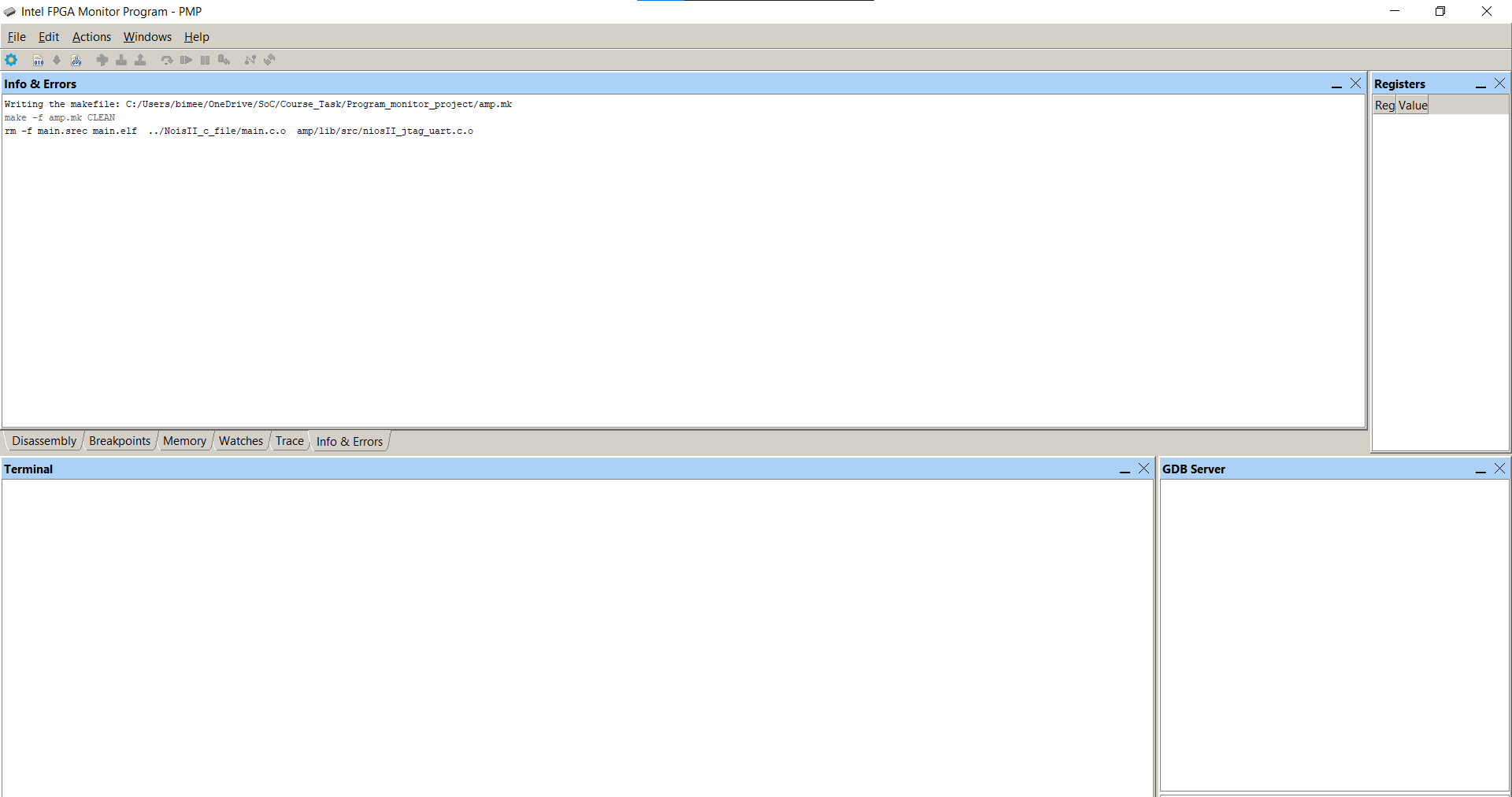


Рис. 2. Програмний термінал в операційній системі «Intel FPGA Monitor Program».

Саме через термінал програмного «Intel FPGA Monitor Program» і буде здійснюватися контроль семисегментних індикаторів.

Такий вибір теми дозволить виконати освітницьку ціль даного курсового проєкту, адже буде реалізована ціла система на кристалі. Вона скрадатиметься з самостійно розробленого модуля, мікропроцесора та об’єднуючих компонентів.

# РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ

## 1.1 Керування семисегментним індикатором.

В даному проекту я використовую семисигментного індикатор, схема якого зображена на Рис. 3.

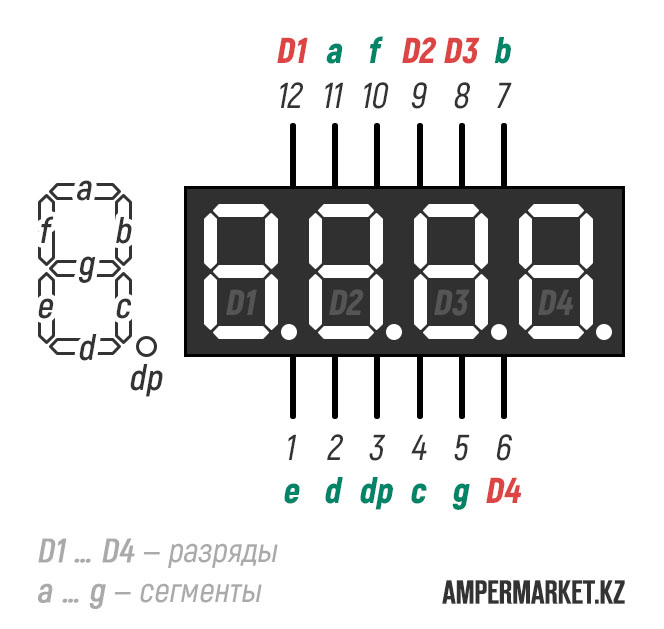


Рис. 3. Схема семисигментного індикатора, що буде використовуватись.

Зауважу, що це індикатор з спільним катодом. Це означає, що для загорання світлодіодів, слід подавати на «логічну 1» на піни a, b, c, d, e, f, g та при цьому гарантувати «логічний 0» на об’єднаних катодах D1, D2, D3, D4.

Для того, щоб відображати різні значення на кожному з індикаторів, слід швидко перемикатись між ними. В моєму проекті задачу контролю семисегментним індикатором на себе бере оригінальний модуль, формований на базі компонентів плати програмованої логіки DE10-nano. Цей модуль буде отримувати 16 біт інформації, які буде відображати на індикаторах в шіснадцятковій системі числення.

## 1.2 Робота з терміналом середовища “Intel FPGA Monitor Program”

«Intel FPGA Monitor Program» – це програмне середовище, що дозволяє працювати з мікропроцесорним ядром NiosII, яке в свою чергу інтегроване в мікросхему системи на кристалі на платі DE10-Nano. Цією мікросхемою є чіп 5CSEBA6U23I7.

Для роботи використовується стандартна бібліотека мови програмування «С» - бібліотека «stdio.h».

Зазначу, що будувати програмне забезпечення слід з врахуванням адреси, яка належатиме модулю керування семисигментним індикатором. Для компілювання можна використовувати те ж середовище «Intel FPGA Monitor Program», от тільки для цього слід встановити систему «Windows Subsystem for Linux», а також добавити деякі «Шляхи» в «Змінні оточення системи Windows».

Тільки після цього, появляється можливість компілювати файл з розширенням «.с». Отриманий бінарний файл з інструкціями для ядра і буде завантажений в модуль пам’яті і стане «пам’яттю програм» для мікропроцесорного ядра. Компіляція зображена на Рис. 4.

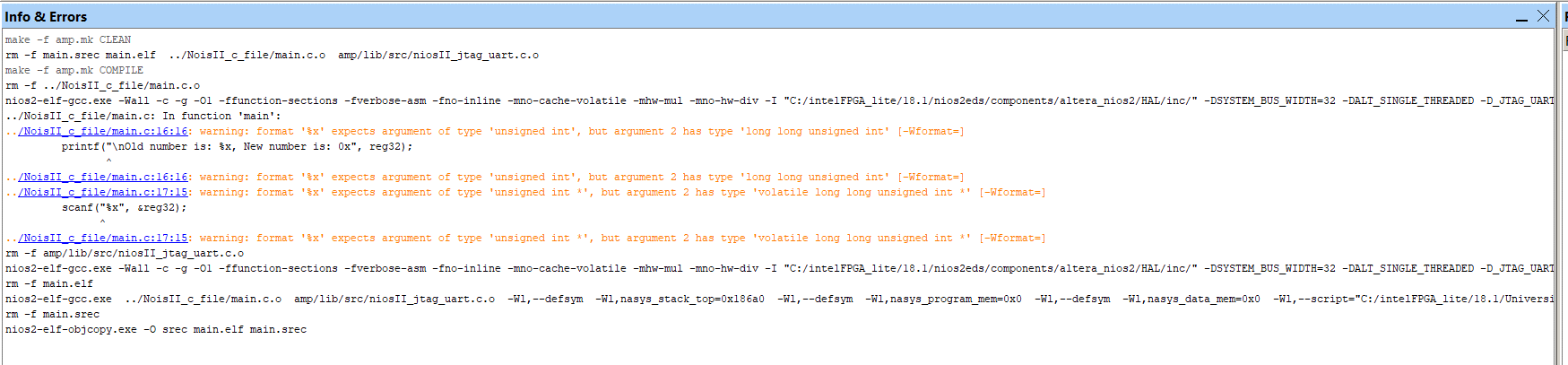


Рис. 4. Компіляція фалу «main.c».

# РОЗДІЛ 2. СТРУКТУРНА СХЕМА ПРИСТРОЮ

## 2.1 Структурна схема пристрою та її опис

На Рис. 5 зображена схематичне представлення пристрою.

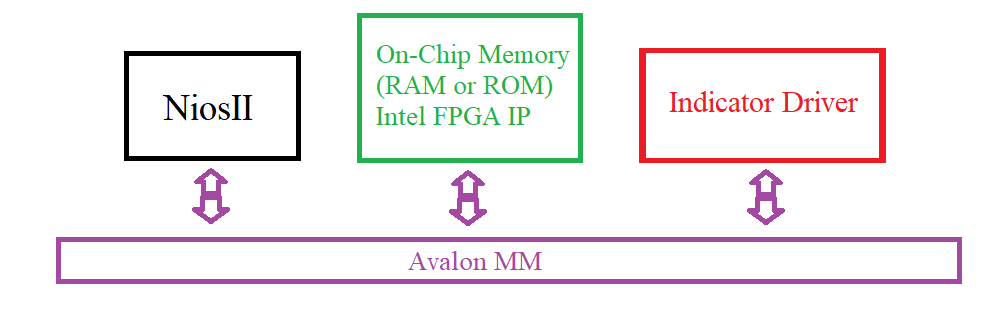
****

Рис. 5. Схематичне зображення системи на кристалі.

Компонент «NiosII» – це вбудоване в систему на кристалі мікропроцесорне ядро.

Компонент «On-Chip Memory (RAM or ROM) Intel FPGA IP» – це блок, що утворюється з компонентів пам’яті всередині системи на кристалі. Він утворюється об’єднанням невеликих комірок в одну більшу систему пам’яті.

Компонент «Indicator Driver» – це оригінальний блок, що HDL описується мовою «verilog». Цей модуль синтезується з різноманітних складових програмованої логіки системи на кристалі. Лістинг коду цього модуля зображено на Рис. 7:

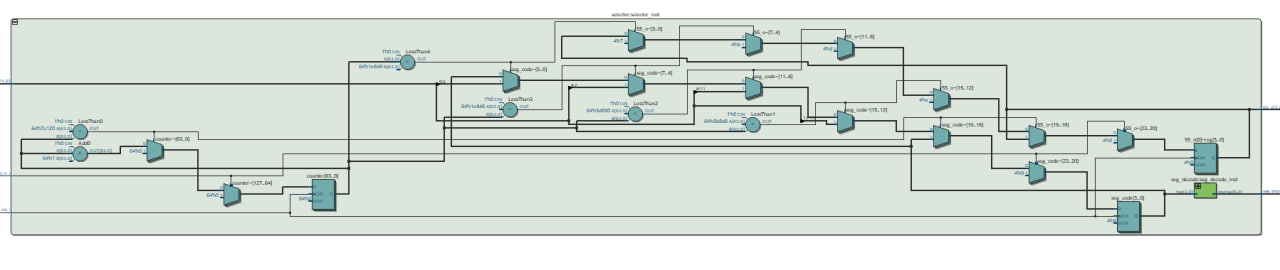


Рис. 6. Вигляд модуля «Indicator Driver» в утиліті «RLT Viewer»

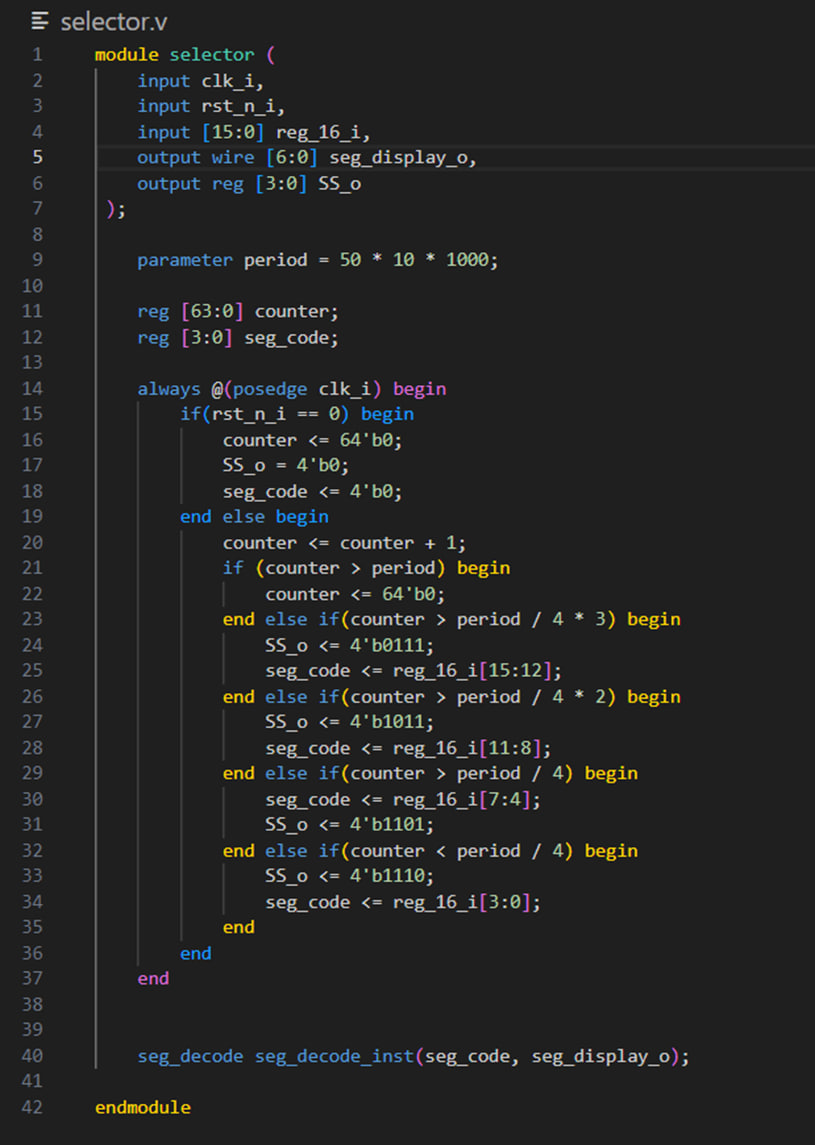


Рис. 7. Лістинг коду на мові «verilog», що описує модуль «Indicator Driver».

# РОЗДІЛ 3. СТВОРЕННЯ ПРИСТРОЮ

## 3.1. Створення та симуляція модуля «Indicator Driver»

Під час написання файлу «selector.v», для того, щоб перевіряти модуль, було написано написано допоміжний файл «tb\_for\_selector.v» (рис. 8) та файл «sim.do» (рис. 9) для запуску симуляції в середовищі «ModelSim».

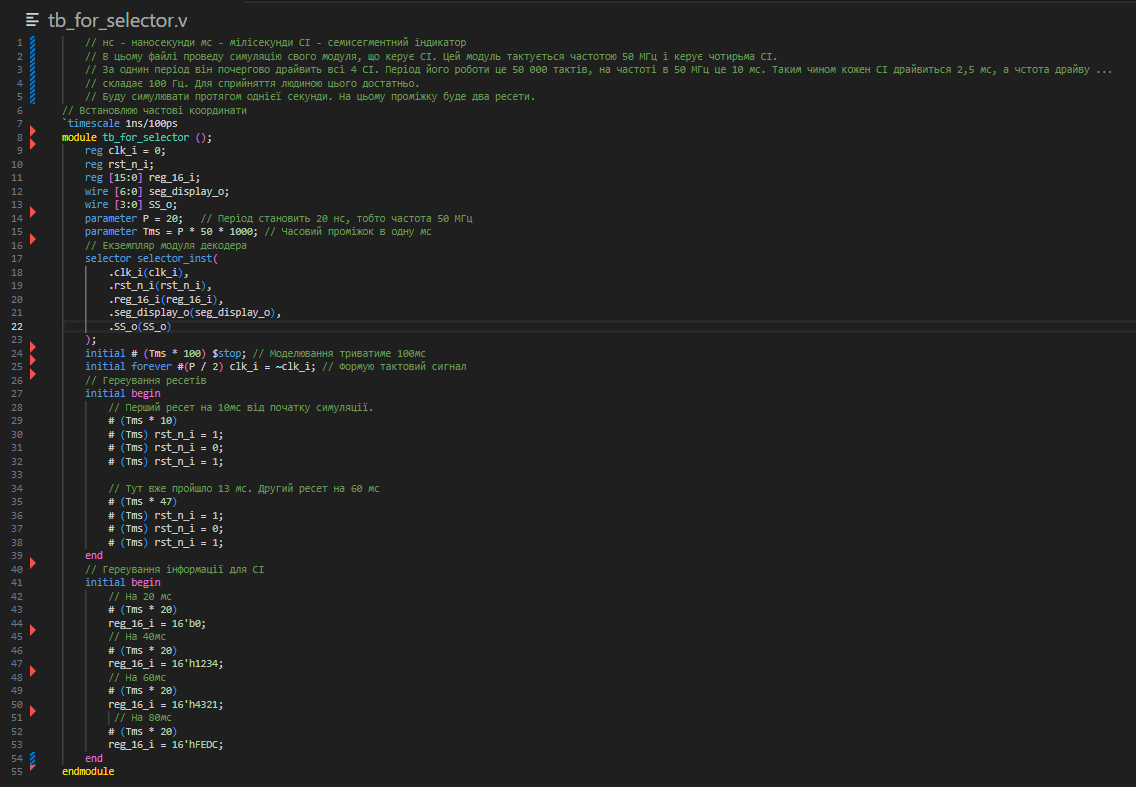


Рис.8. Лістинг файлу «tb\_for\_selector.v»

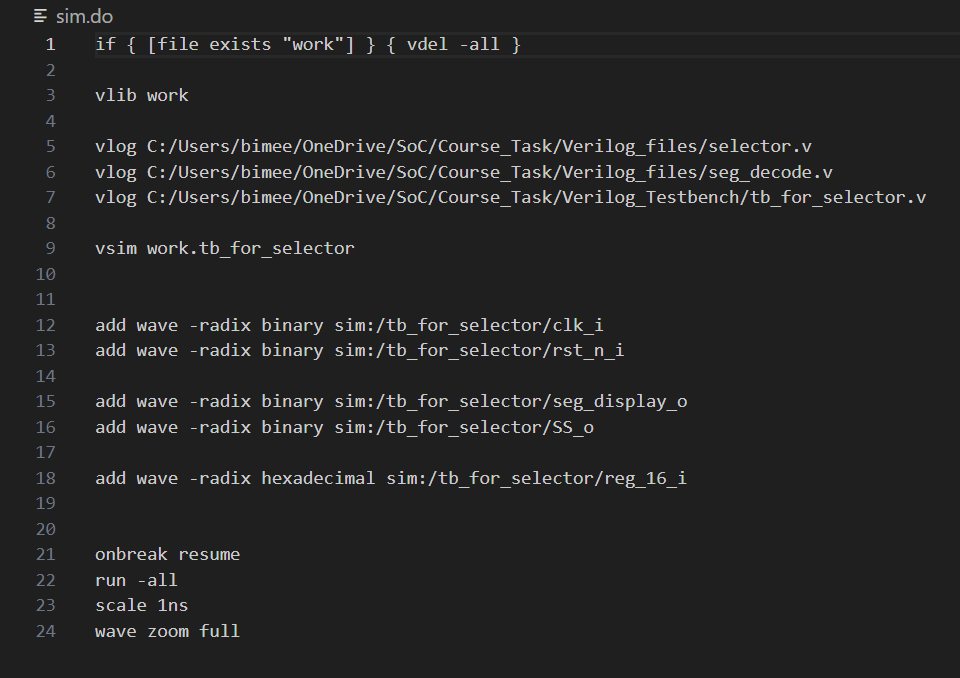


Рис.9. Лістинг файлу «sim.do»

Запустивши симуляцію командою «vsim -do sim.do» можна побачити часову діаграму зображену на рис. 10.

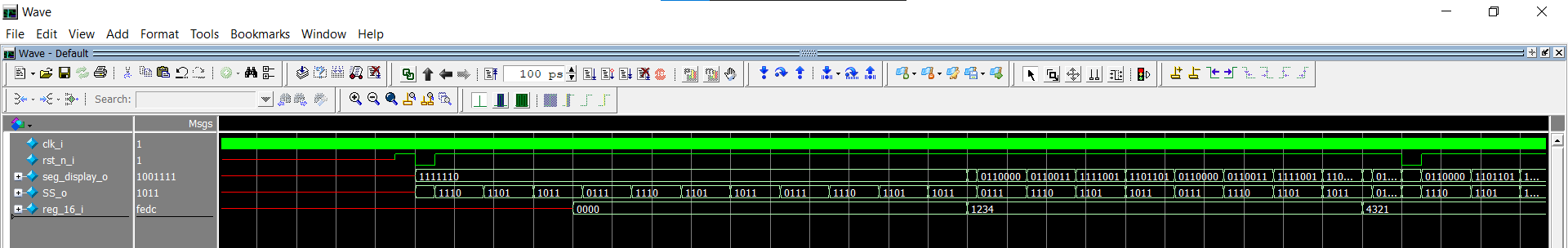


Рис. 10. Симуляція роботи модуля «selector», який і є компонентом «Indicator Driver».

Зауважу, що кожен біт «Slave select» є катодом і тому світиться саме той індикатор чий «Slave select» рівний нулю, тому що саме тоді через світлодіоди протікає струм. Щоб зрозуміти якому біту відповідає який світлодіод, дивіться рис. 11.

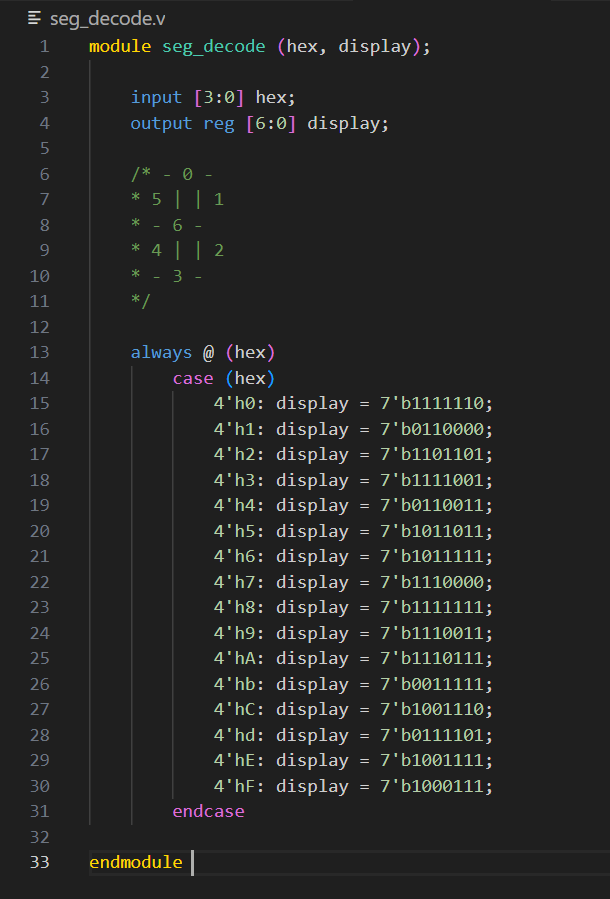


Рис.11. Лістинг файлу «seg\_decode.v»

## 3.2. Проєкт в середовищі «Quartus»

При створенні проєкту в середовищі «Quartus», першим ділом було додано файли як на Рис. 12.

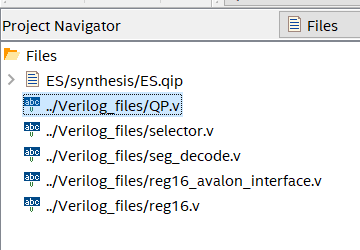


Рис.12. Файли проєкту в середовищі «Quartus».

Файлом верхнього рівня ієрархії є файл «QP.v» (Рис. 13), тобто «Quartus Project». Він містить мій оригінальний модуль «selector» та модуль, що описується в файлі «ES.qip» (Рис. 14).

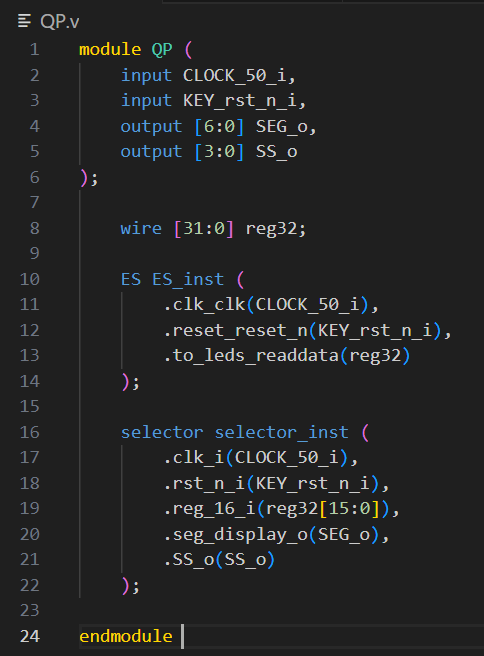


Рис. 13. Файл «QP.v».

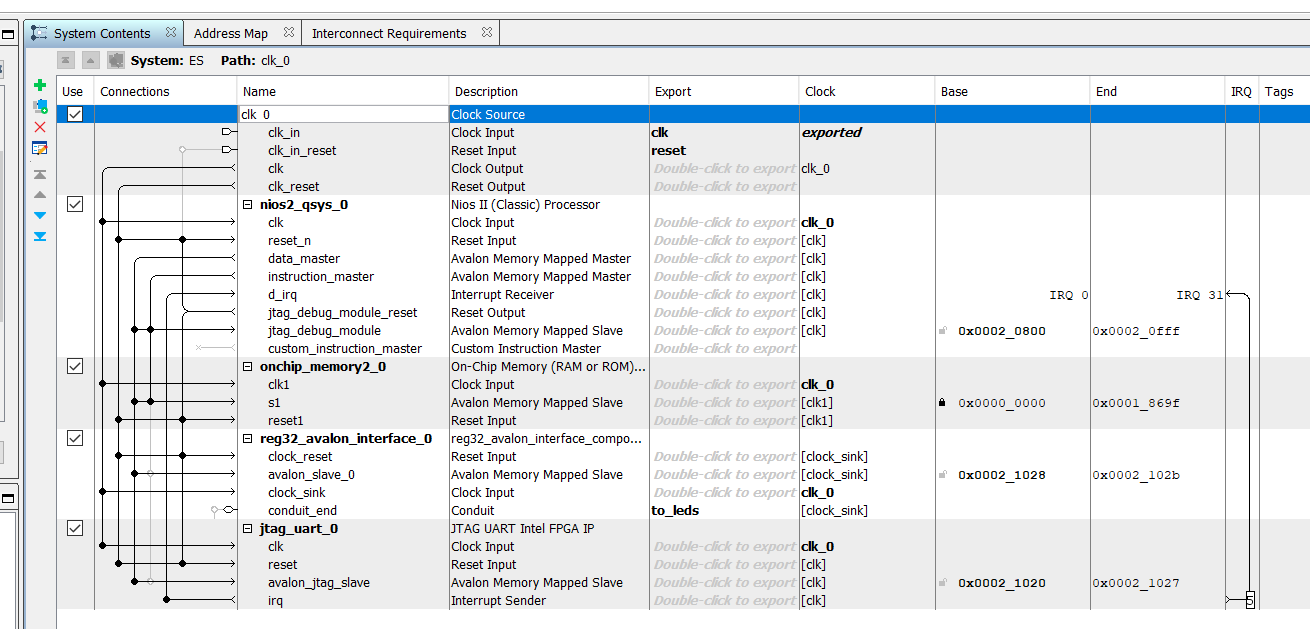


Рис. 14. Файл «ES.qip».

Як видно з рис. 14, модуль ES (Embedded System) є файлом конфігурації системи на кристалів для мікросхеми FPGA. Саме в цьому модулі компоненти і були налаштовані, з’єднані шиною «Avalon MM» та були встановлені їх адреси.

Файл «reg32.v» (рис. 15) – виступає в ролі 4 байт пам’яті яка спільна для ядра та оригінального модуля. Було би достатньо і 16 біт, але оскільки шина даних «Avalon MM» має 32 біти, то робити модуль меншим за 32 немає сенсу.

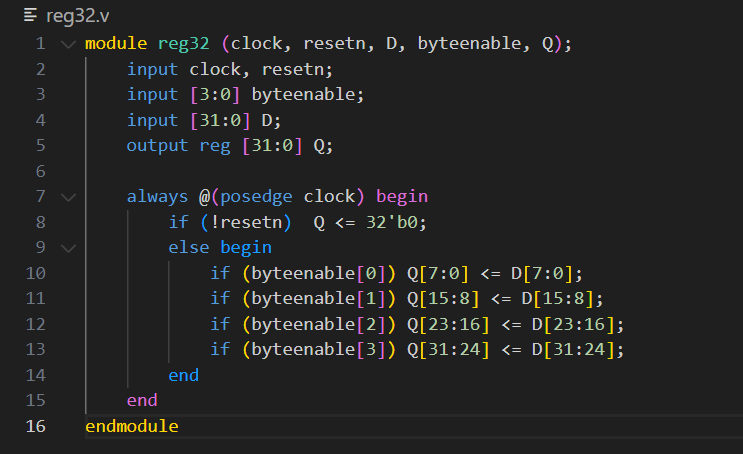


Рис. 15. Файл «reg32.v»

Файл «reg32\_avalon\_interface.v» (рис. 16) – є файлом, що забезпечує необхідні виходи для підключення шини «Avalon MM» до модуля «reg32». Електричні схеми обох модулів зображені на рис. 17.

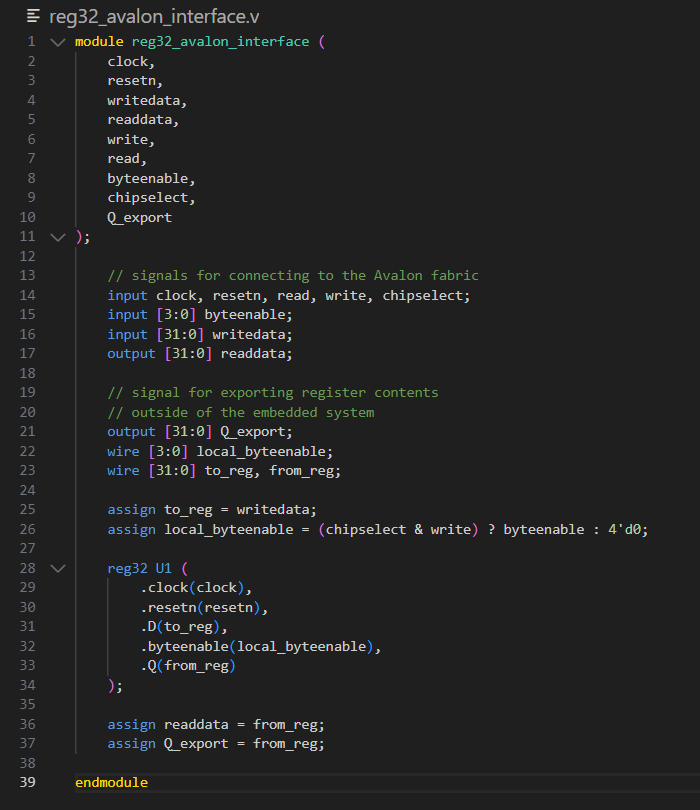


Рис. 16. Файл «reg32\_avalon\_interface.v»

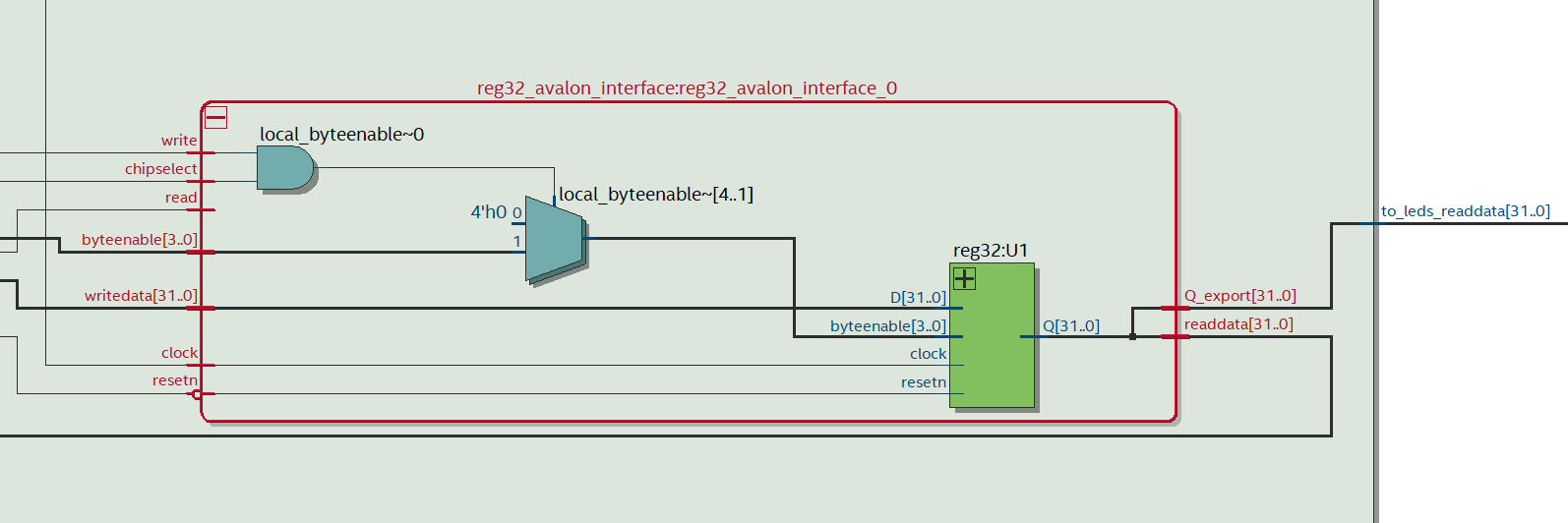


Рис. 17. Електричні схеми модулів «reg32\_avalon\_interface.v» та «reg32.v»

## 3.3. Проєкт в середовищі «Intel FPGA Monitor Program»

Для того, щоб завантажити код програми для процесора NoisII слід створити проєкт в середовищі «Intel FPGA Monitor Program». Ось якими параметрами він володіє (рис. 18):

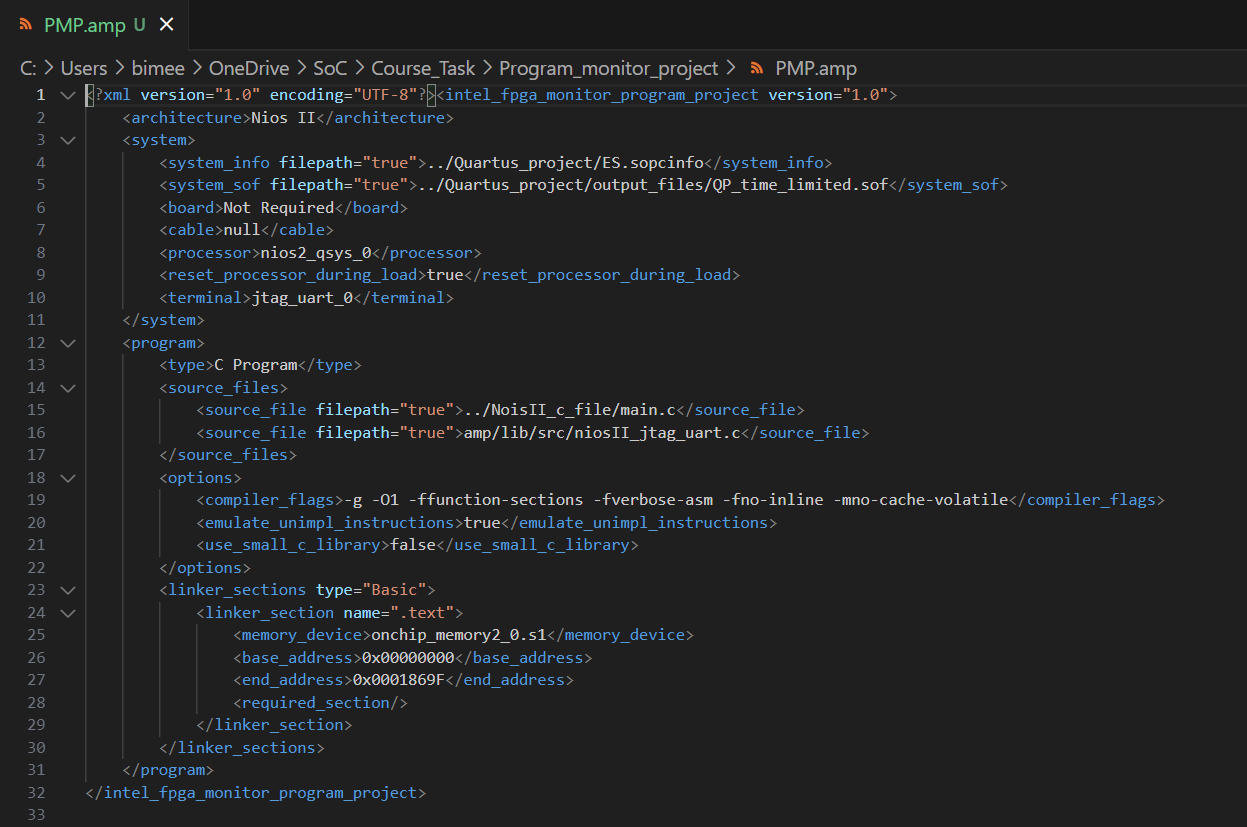


Рис. 18. Файл «PMP.amp», що і визначає налаштування проєкту.

А ось і алгоритм роботи процесора (рис. 19):



Рис. 19. Файл «main.c», що задає алгоритм роботи процесору.

Оскільки в мене під час виконання курсового проєкту виникли проблеми з платою DE10-nano (рис. 20), то я дещо зміню «main.c» та продемонструю його роботу у відриві від решти компонентів системи (рис. 21).

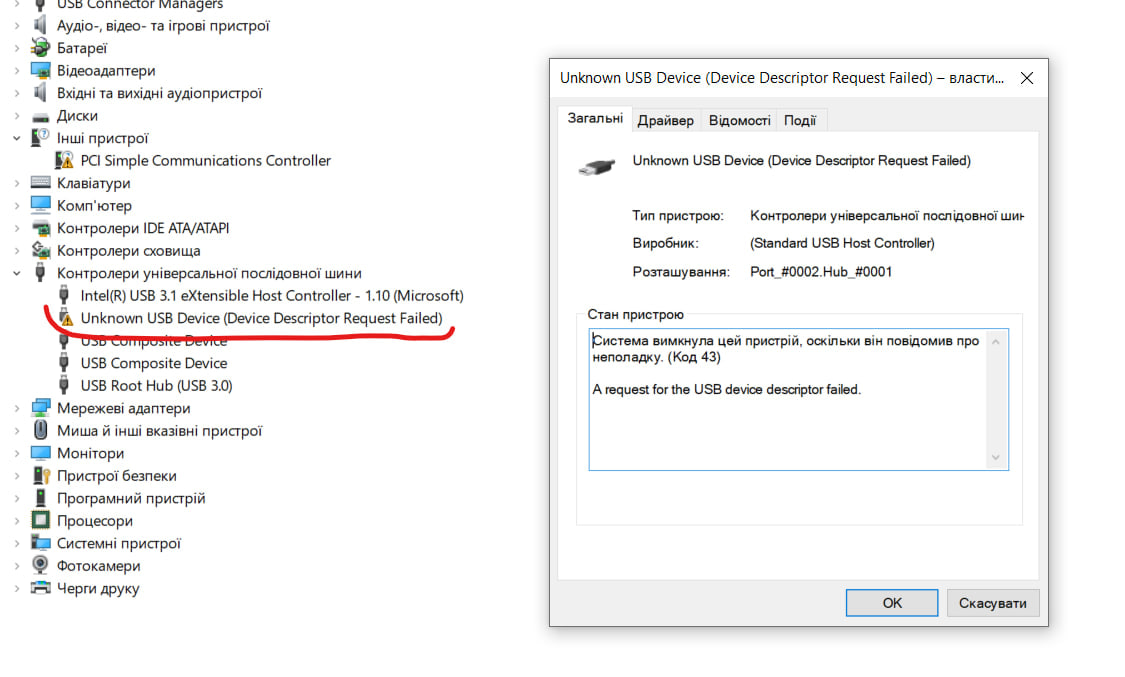


Рис. 20. Вихід з ладу плати DE10-nano.

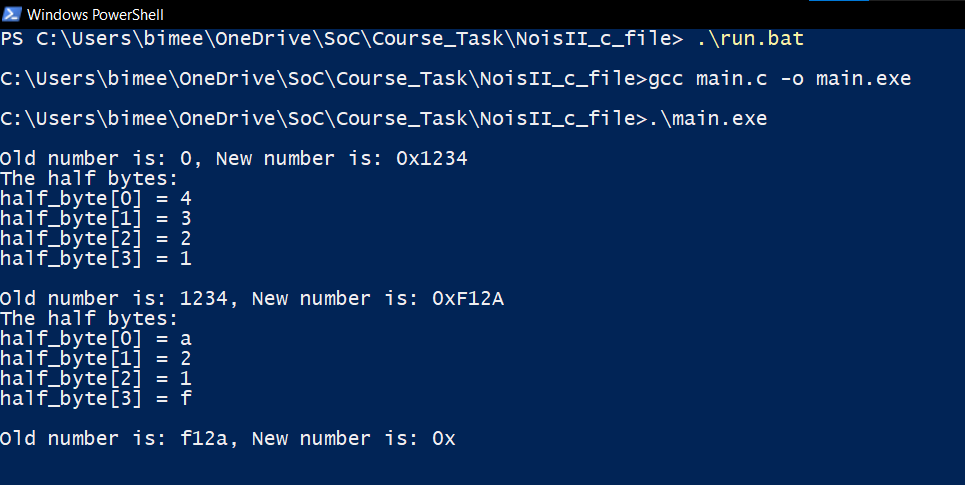


Рис. 21. Демонстрація роботи «main.c» у відриві від решти компонентів системи на кристалі.

# ВИСНОВОК

В межах даного курсового проєкту було розроблено систему на кристалі, котра базується на платі DE10-nano. Ця система використовує взаємодію процесорне ядро та засоби відладки для того, щоб обробляти інформацію, що надходить з терміналу утиліти «Intel FPGA Monitor Program». Також є оригінальний компонент, що отримує дані від ядра через шину «Avalon MM» та у відповідності до отриманої інформації керує сукупністю семисигментних індикаторів. Система містить й інші компоненти як от пам'ять чи порти PIO.

Нажаль, мені не вдалось перевірити всі елементи в взаємодії та добавити такий експеримент в пояснювальну записку, але враховуючи досвід запуску подібних систем в межах іншої дисципліни, вважаю, що пристрій розроблено без грубих помилок. Також присутня перевірки оригінального модуля за допомогою засобу симуляції «ModelSim» та перевірка програмного забезпечення процесора NiosII у відриві від решти компонентів.

Враховуючи такий результат, вважаю, що курсовий проєкт заслуговую оцінки «достатньо»!

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. DE10-Nano User Manual [електроний ресурс] – Режим доступу: <https://ftp.intel.com/Public/Pub/fpgaup/pub/Intel_Material/Boards/DE10-Nano/DE10_Nano_User_Manual.pdf>
2. DE10-Nano Board [електроний ресурс] – Режим доступу: <https://www.mikrocontroller.net/attachment/327098/de10-nano_a0.pdf>
3. Introduction to the Intel Nios II Soft Processor [електроний ресурс] – Режим доступу: <https://classroom.google.com/c/ODU2NzY2MzUyMjFa/m/MTY5MTMxNTg0NTcx/details>