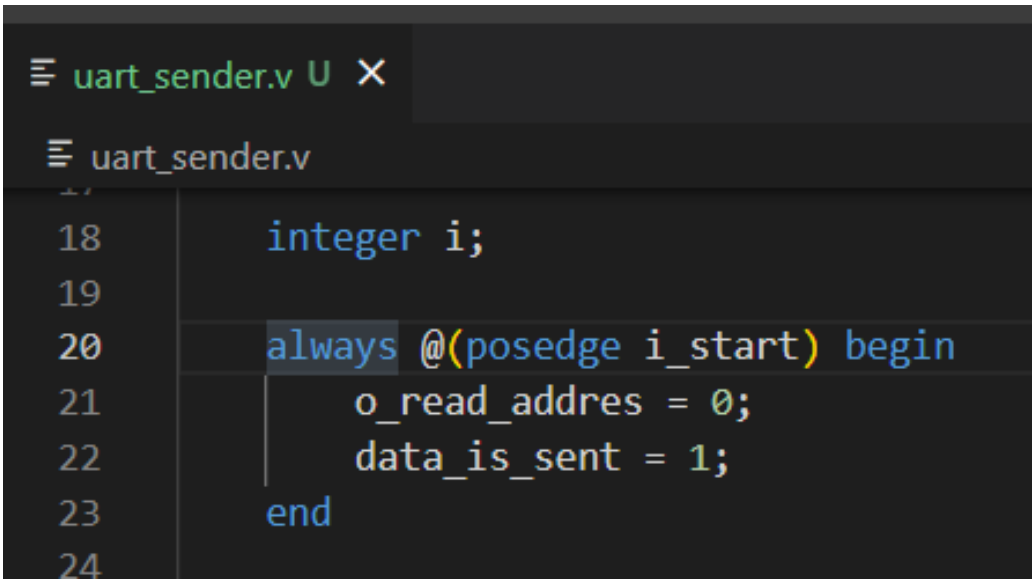
Доопрацювати процесорне ядро MIPS та долучити до нього наступний периферійний пристрій

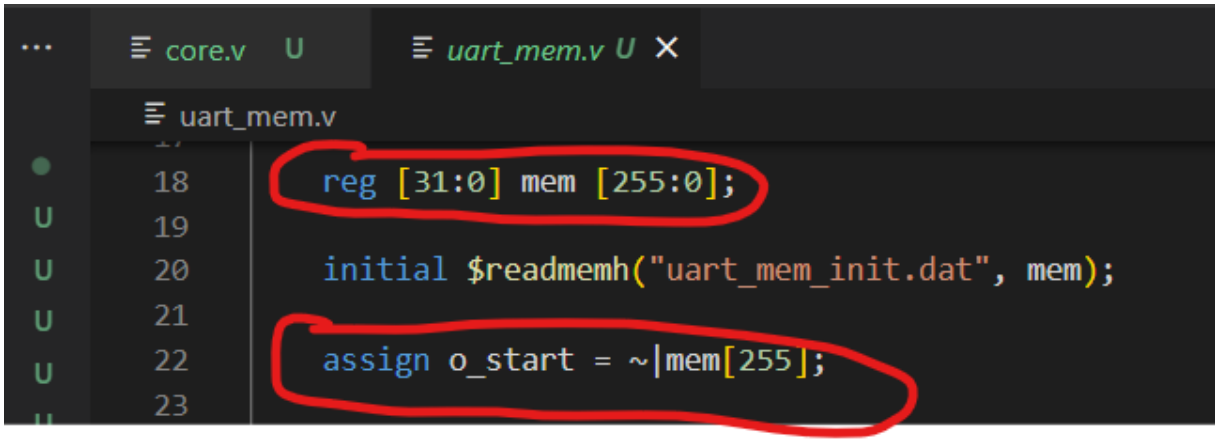
№	ДК-91	Тема розрахункової графічної роботи
1		Контроллер формування керуючих сигналів зовнішнім пристроєм
2		Контроллер генератору сигналів з протоколом 1-Wire
3		Контроллер передачі атмосферних параметрів протоколом SPI
4		Контроллер передачі власних даних протоколом I2C
5		Контроллер передачі числових даних протоколом UART
6		Контроллер для передачі власних даних протоколом SPI
7		Контроллер передачі синусоїдального сигналу до ЦАП шиною UART
8		Контроллер формування ШИМ-сигналу по шині SPI
9		Контроллер передачі числових даних протоколом SPI для відображення на індикаторах
10		Контроллер генератору сигналів з протоколом I2C

Алгоритм:

- Адреса uart_mem починається з 32 і йде до 287 (255 + 32).
- Щоб почати надсилання потрібно взвести сигнал start



- Для цього потрібно в остайню комірку памяті контролера записати нуль



- Використовую таку команду

```
tb_sim.v U
tb_sim.v
114   */
115
116   /*
117           RGR
118       ac00011a - Op = 43| Rs = 0 | Rt = 0 | Imm = 287
119   */
```

- Це єдина команда яка є в памяті

```
tb_sim.v U    rom_init.dat U
rom_init.dat
1    ac00011f
```

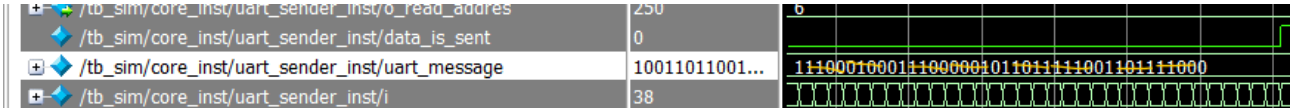
- Після того як виконалась ця команда почнеться передача по юарту
- Спершу встановиться сигнал start
- Потім sender почне починабчи з нуля збільшувати значення на адресі читання uart_mem
- При цьому sender буде діставати по 32 біти данних, тобто по 4 байти
- Передаватиме він по юарту кадри з розміром корисної інформації 8 біт
- Гармонічний сигнал я створив за допомогою сайту [Sine Look Up Table Generator Calculator (daycounter.com)](daycounter.com) та записав його в пам'ять контролера

```
19
20      initial $readmemh("uart_mem_init.dat", mem);
```

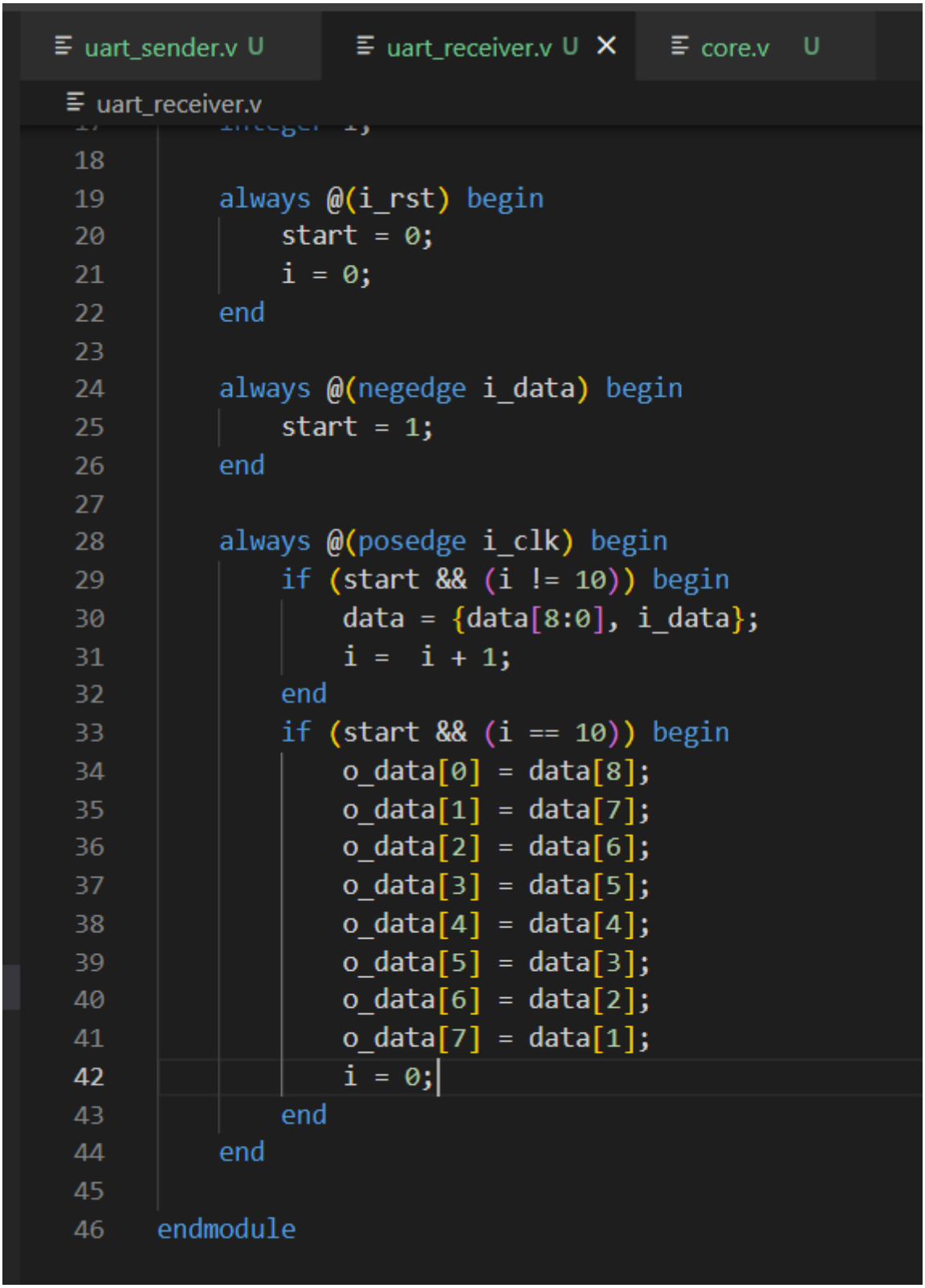
- Сигнал юарт формується наступним чином

```
28      uart_message = { 1'b1, i_data[7:0], 1'b0, 1'b1, i_data[15:8], 1'b0, 1'b1, i_data[23:16], 1'b0, 1'b1, i_data[31:24], 1'b0 };
```

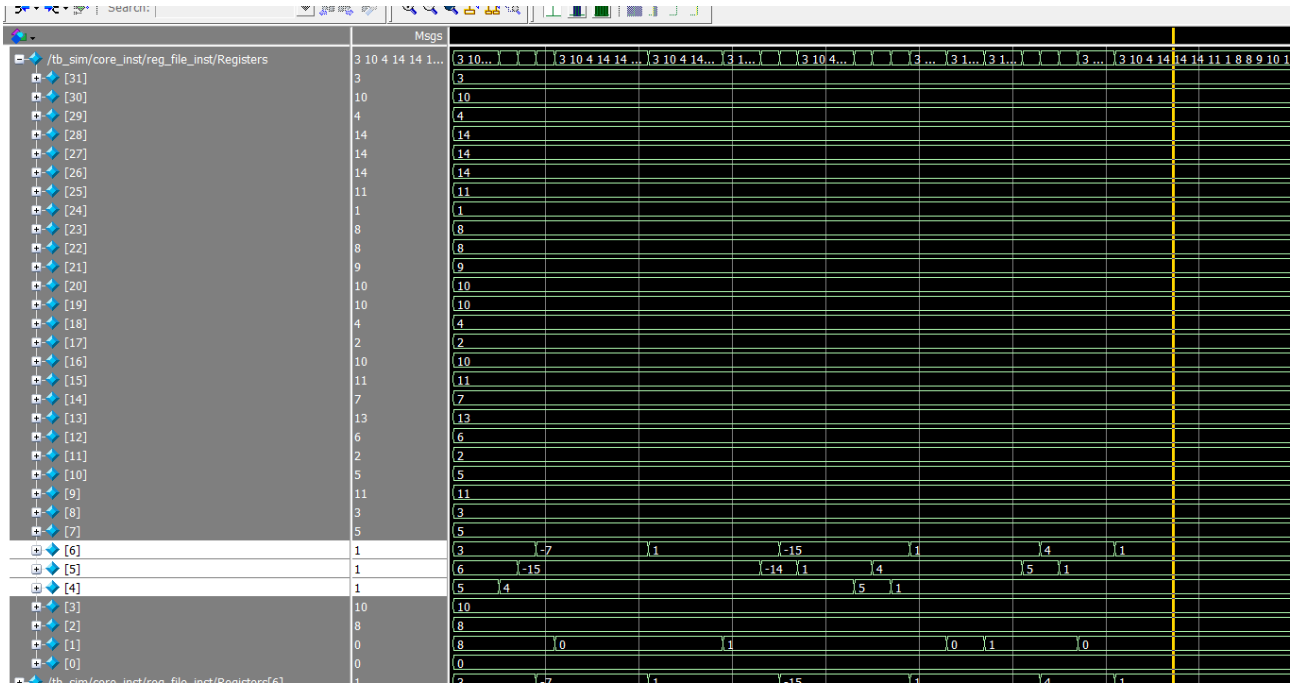
- А ось який сигнал передається юартом (зачеркнув корисні біти)



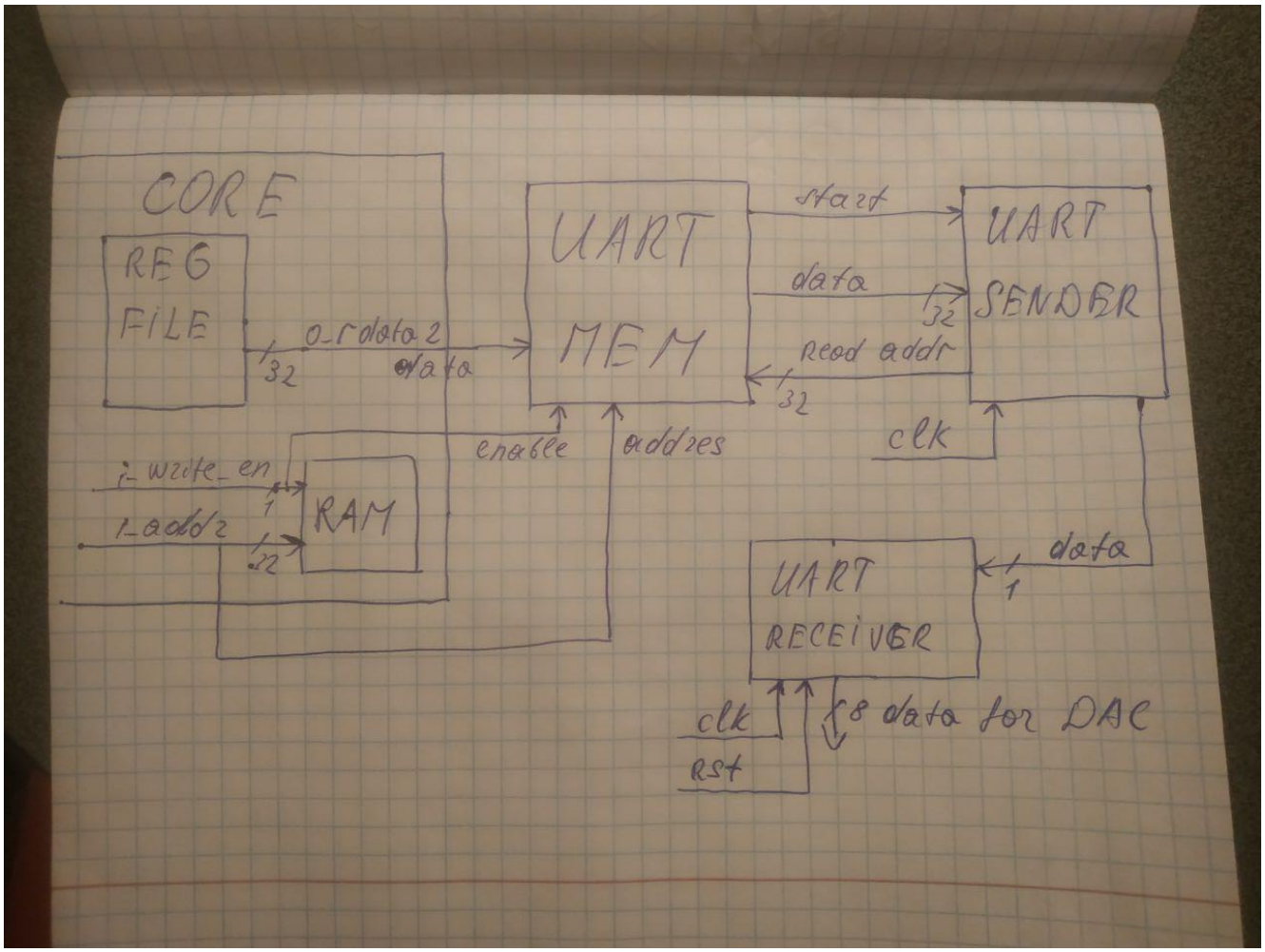
- Ресівер ж приймає сигнал за рахунок чіткої синхронізації з сендером
- Негативний фронт означає старт біт і його ловить і посчинає прийом данних



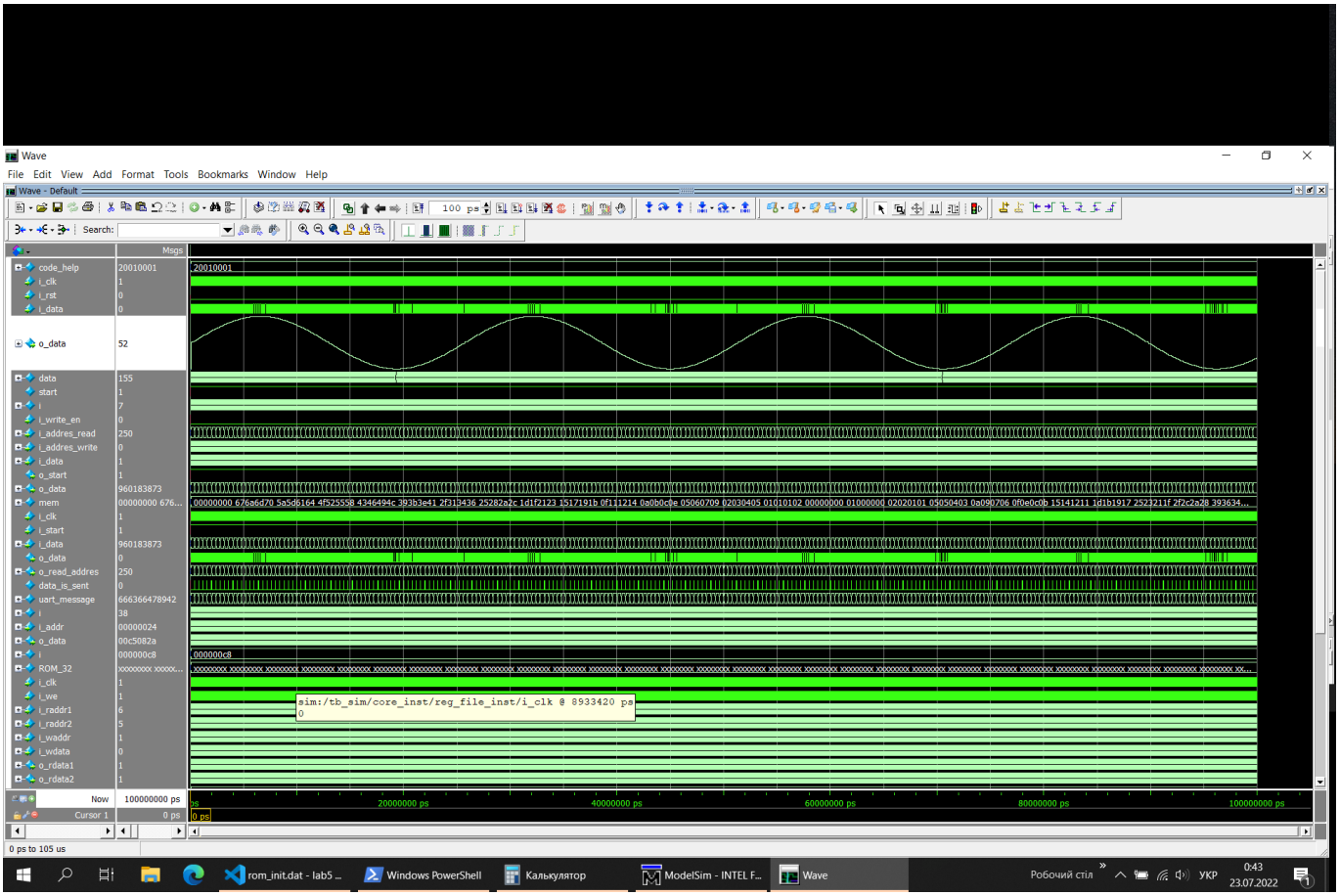
- Також є ресет але для його використання слід знаново обновити остайню комірку памяті котролерра
- Зауважу що сам поцесор не задіяий в передачі данних і може спокійно виконувати іншу роботу
- Ось наприклад аиконання сортування паралельно з ЛР 5 паралельно роботі контролерра



- Загальна схема виглядає так



- Сигнал на виході ресівера виглядає так



- Для перевірки найпростіше запустити проект командою консолі vsim –do sim.do