### 0. Algorytm - wzór, cel.

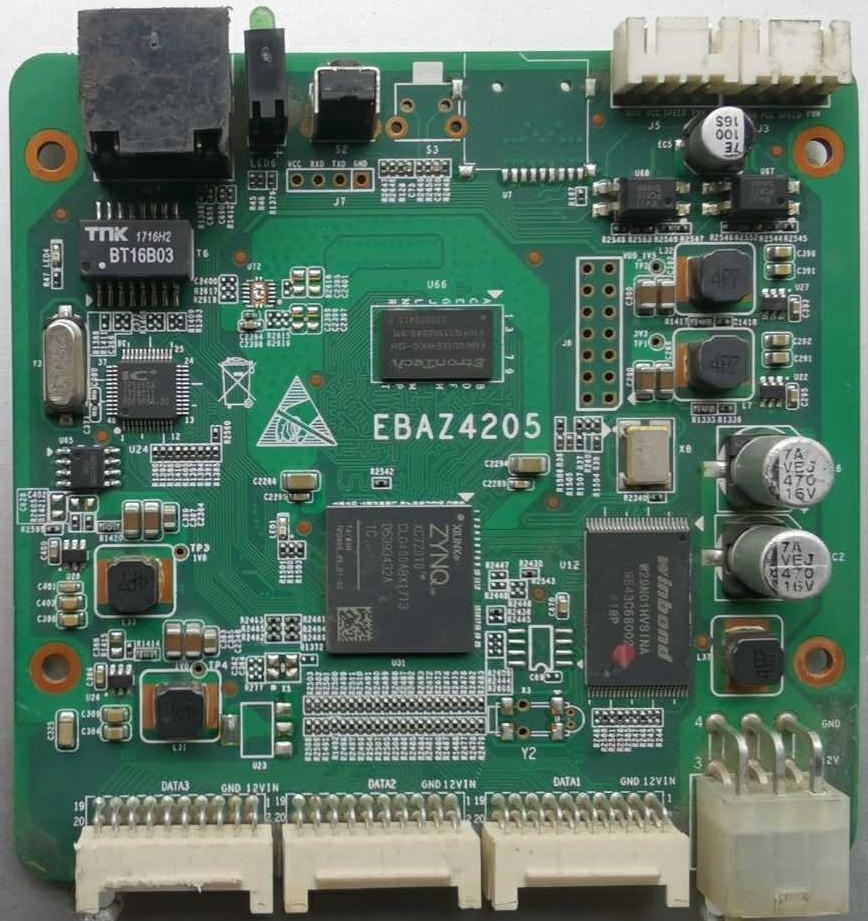
Celem projektu jest stworzenie sumatora szeregowego w postaci akceleratora sprzętowego mikroprocesora Zynq.

Projekt polega na realizacji procesora który na wejściu otrzymuje równocześnie M bitów (np. 32 bity). Są to pojedyncze bity M liczb, których kolejne bity b(0), b(1), b(2), b(3), ..., b(N-1) są podawane na wejście procesora w kolejnych przesłaniach z CPU. Zadaniem bloku dedykowanego jest sumowanie liczb w kolejnych krokach i udostępnienie CPU do odczytu wyniku sumy.

Plan rozwoju projektu:

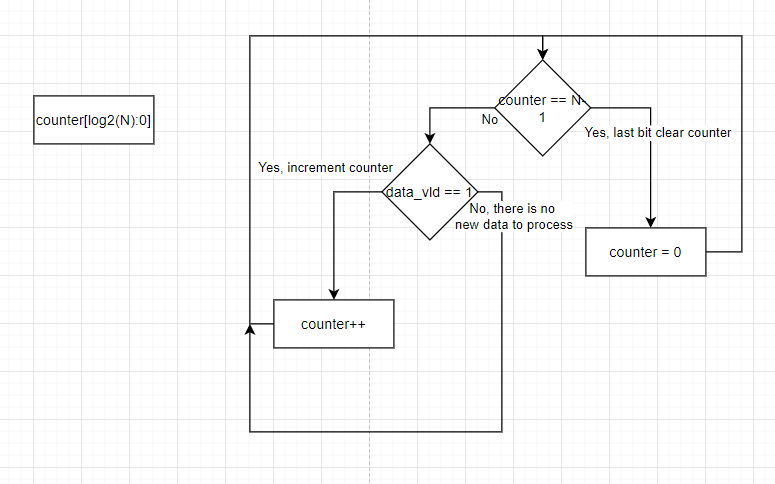
* Podczas projektu powstaną co najmniej 2 wersje sumatora (jedna stworzona przez jedną osobę i druga przez drugą), ma to na celu sprawdzenie różnych sposobów implementacji oraz ich późniejsze porównanie (np pod względem opóźnienia/ szybkości działania, zajętych zasobów itp.)
* Wybrana wersja sumatora zostanie szczegółowo opisana zgodnie z planem projektu zaproponowanym na zajęciach,
* Na koniec stworzona zostanie wersja potokowa tego akceleratora sprzętowego (razem w dwie osoby)

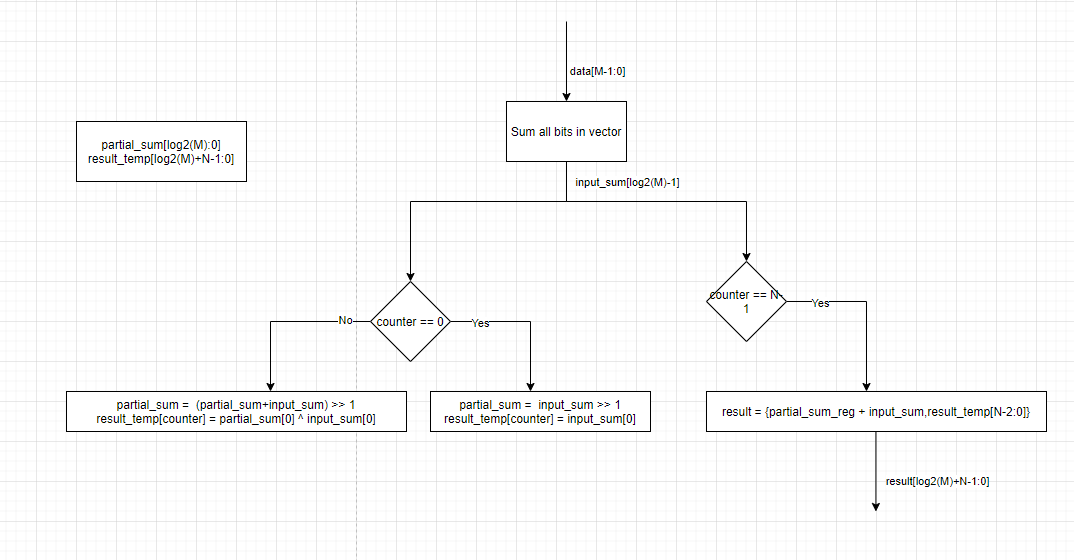
Wszystkie testy zaprojektowanych akceleratorów zostaną przeprowadzone na płytce **EBAZ4205** z układem **ZynQ XC7Z010**

****

Wyniki działania zaprojektowanego akceleratora sprzętowego będą przesyłane poprzez wbudowany interfejs UART w celu wyświetlenia w konsoli.

### 1. Rozpisanie algorytmu w postaci diagramu.





### 2. Przepisanie algorytmu behawioralnie w Verilogu/SV.

### 3. Stworzenie modułu, określenie wejść/wyjść, sposobu dostępu do pamięci.

### 4. Testbench potwierdzający prawidłowe działanie modułu - wektory wczytywane z pliku, zapis wyników do pliku.

### 5. Przepisanie moduły na wersję syntezowalną.

### 6. Testbench z punktu 4 na module z 5 - wyniki.

### 7. Uzbrojenie modułu w AXI(lite), podłączenie do procesora uB, prosty driver, symulacja.

### 8. Uruchomienie systemu na sprzęcie (ARM).

### \*9. Wersja potokowa modułu, symulacja, implementacja na sprzęt.

TODO:

* tabelka jak parametry N i M wpływają na zasoby FPGA
* Bardzo długie liczby np 1024 bity sprawdzić
* Wynik ma być gotowy od razu nawet jeśli się przerwie podczas przesyłania
* Bit T last
* Vivado duzo ip coreów, fifo do wyjścia, jest wizard, wejscie 1 bit i wyjscie wiele bit
* Opis przebiegów dobry
* Symulacja post timingowa
* Pomiar czasu timerem / oscyloskopem, fit interval timer.
* Umiescic ten dokument na githubie