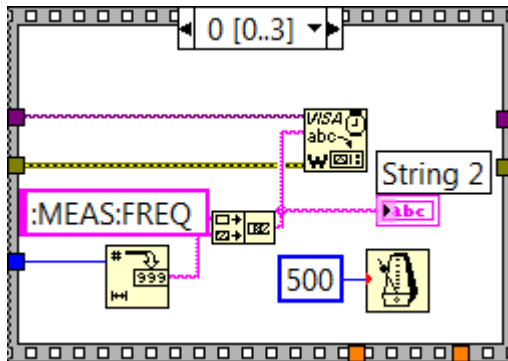


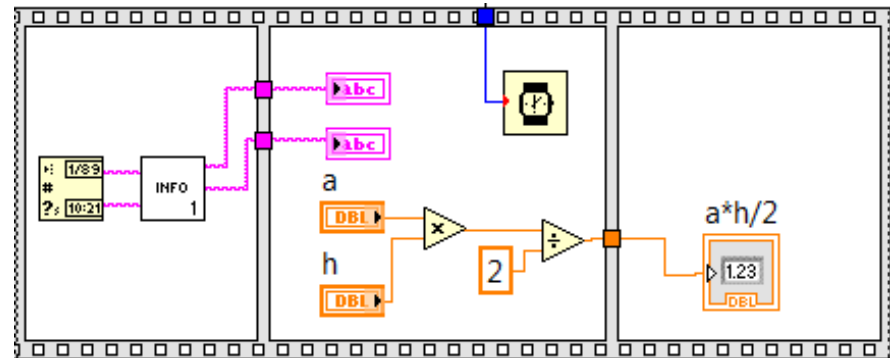
Konstrukcje sekwencyjne. Zmienne lokalne.

- Konstrukcje sekwencyjne pozwalają na wymuszenia określonej kolejności wykonywania fragmentów kodu, gdy nie daje się tego uzyskać przepływem danych.
- Konstrukcja sekwencyjna wygląda jak ramka filmu i składa się z jednej lub kilku ramek ułożonych w stos (*stacked sequence*) lub płasko jedna za drugą (*flat sequence*) .
- Oba rodzaje konstrukcji działają podobnie ale posiadają nieco odmienne właściwości.



Stacked sequence:

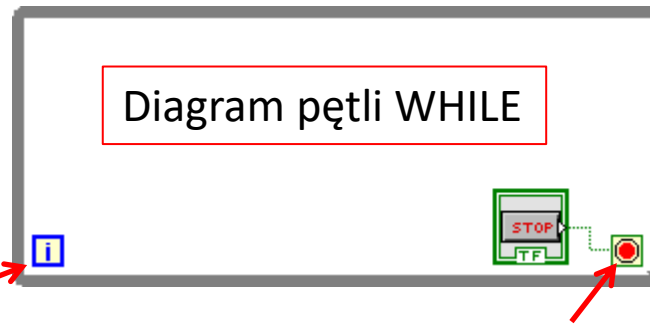
Cztery ramki w układzie stosu.



Flat sequence:

Trzy ramki w układzie płaskim.

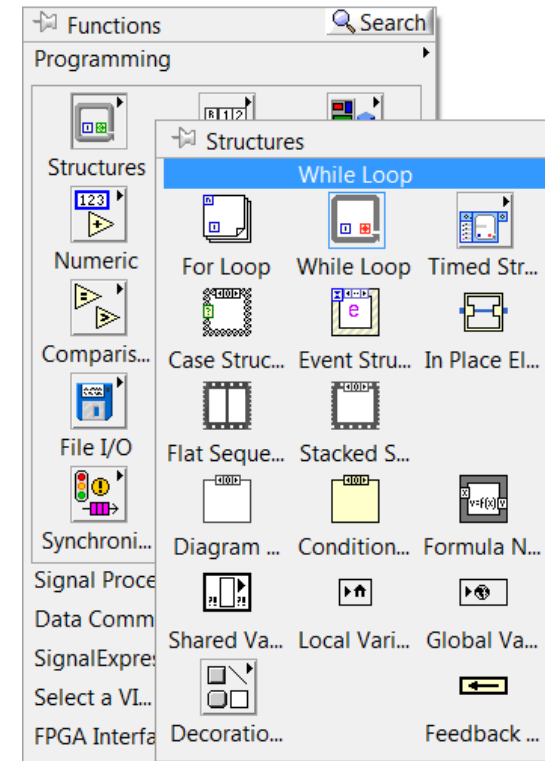
Pętla WHILE, działanie pętli WHILE



Licznik iteracji

Terminal warunku działania pętli

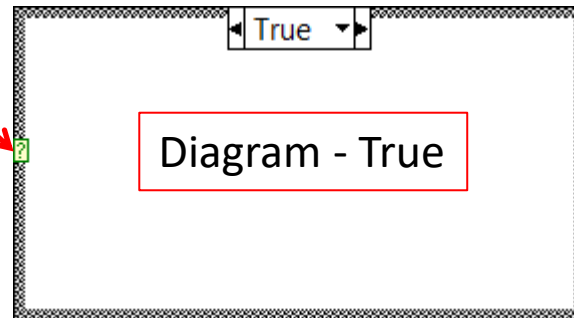
- Konstrukcję pętli **while** stosuje się w celu cyklicznego wykonania wybranego bloku kodu, gdy liczba wymaganych powtórzeń (iteracji) wykonania nie jest znana.
- Ramka pętli **while** posiada predefiniowany **terminal wyjścia licznika iteracji** oraz **terminal warunku kontynuowania działania pętli**.
- Licznik iteracji jest zerowany w momencie rozpoczęcia działania pętli, dostarcza aktualny numer wykonywanej iteracji (od 0) i jest inkrementowany po każdej iteracji.
- Terminal kontynuacji działania pętli korzysta z wartości logicznych (boolowskich) wypracowanych przez diagram pętli. Konfigurowanie terminala kontynuacji pozwala ustalić wartość logiczną przerywającą działanie pętli (**Continue IF True** lub **Stop If True**).
- Sprawdzenie warunku zakończenia jest realizowane po wykonaniu każdej iteracji, dlatego zawsze jest wykonana przynajmniej jedna iteracja pętli.



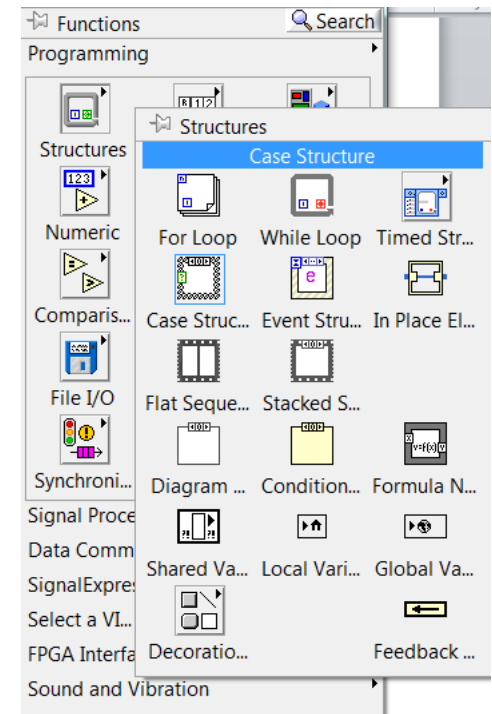
Struktura CASE

Wejście selekcyjne:

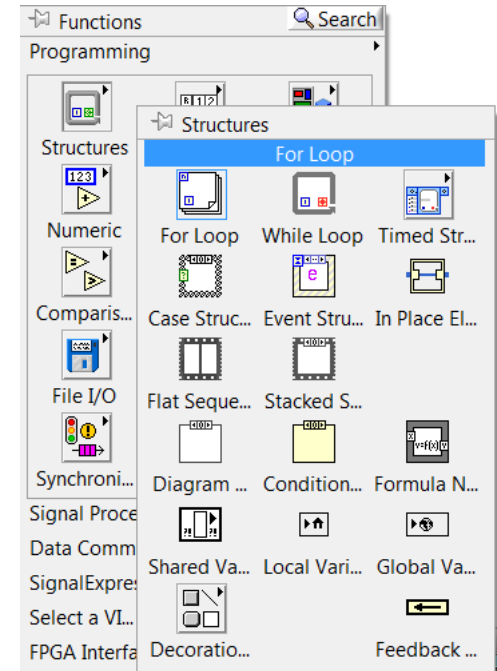
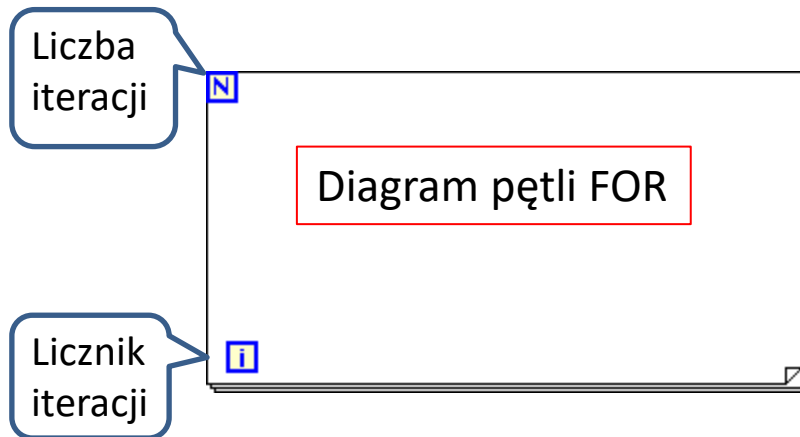
- boolowskie,
- całkowite,
- string.



- Konstrukcja sterująca **case** umożliwia alternatywne wykonywanie bloków kodu objętych tą konstrukcją.
- Funkcjonalnie odpowiada instrukcji **if...then...else** lub **switch** języka C. Konstrukcja posiada minimum dwie ramki.
- Każda ramka zawiera blok programowy realizujący określone operacje oraz deklarację wartości wybierających.
- Wykonanie konstrukcji polega na wykonaniu **kodu tylko jednej z jej ramek**.
- Wybór ramki jest realizowany na podstawie danej dostarczonej do wejścia selekcyjnego konstrukcji **case** (?).
- Dla każdej z możliwych wartości selektora musi być przypisana jedna z ramek konstrukcji **case**.

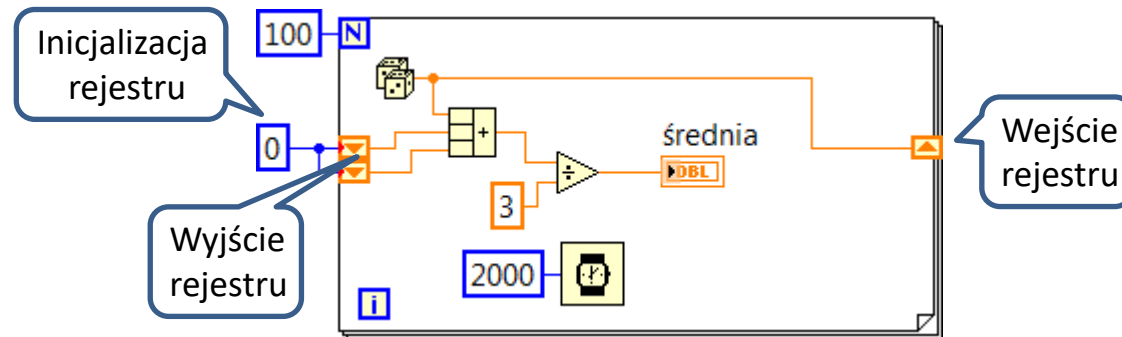


Pętla FOR



- Konstrukcję pętli **for** stosuje się w celu cyklicznego wykonania wybranego bloku kodu, gdy liczba wymaganych powtórzeń (iteracji) wykonania jest znana.
- Ramka pętli for posiada predefiniowane wejście liczby iteracji oraz terminal wyjścia licznika iteracji.
- Wejście liczby iteracji jest typu **long integer**.
- Licznik iteracji jest zerowany w momencie rozpoczęcia działania pętli, dostarcza aktualny numer wykonywanej iteracji (**od 0 do N-1**) i jest inkrementowany po każdej iteracji.
- Sprawdzenie warunku zakończenia (**$i < N$**) jest wykonywane przed rozpoczęciem kolejnej iteracji.

Rejestr przesuwany pętli (*shift registers*)



- Rejestr może być jedno lub wieloelementowy. Każdy rejestr przesuwany pętli pozwala przekazać daną do następnej iteracji pętli.
- Iteracja rozpoczyna się pobraniem danej z rejestru, a kończy wpisaniem do rejestru wyniku działania.
- Następną iteracją pobiera daną z rejestru, czyli korzysta z wyniku uzyskanego w poprzedniej iteracji.
- Rejestr tworzy się wybierając z menu konstrukcji pętli **FOR** lub **WHILE** pozycję **Add Shift Register**. Pętla może mieć wiele rejestrów przesuwanych.
- Terminal ze strzałką skierowaną w górę jest wejściem rejestru.
- Wyjściem rejestru jest terminal ze strzałką skierowaną w dół. Dostarcza on danej z poprzedniej iteracji lub wartość początkową w pierwszej iteracji.

Zad_3.1

Napisz program, który przy użyciu pętli While co 5 sekund generuje przypadkową liczbę całkowitą z zakresu <0,100>. Znajdź różnicę pomiędzy funkcją Wait a funkcją Wait Until Next.

Zad_3.2

Zmodyfikuj program z Zad_3.1 tak aby wykonywał jedynie określoną liczbę powtórzeń (iteracji).

Zad_3.3

W programie z Zad_3.2 dokonaj zamiany pętli While na pętlę For. Następnie uzupełnij program o procedurę obliczającą 10 pierwszych wyrazów ciągu Fibonacciego.