

Дәріс 4. LabVIEW ортасында for және while циклдері

Мақсаты: Студенттерге For Loops және While циклдерінің LabVIEW ішіндегі кейбір негізгі функциялары мен қолданылуын және оларды пайдаланудың сәйкес уақыттарын үйрету.

Кіріспе

For Loop цикл ішіндегі кодтың ішкі диаграмма деп аталатын бөлігін бірнеше рет орындайды. Санау N терминалы ішкі диаграмманы қайталау санын орнатады. Ішкі диаграмманың бір қайталануы итерация деп аталады.

For Loop функциясын әрекеттің белгілі бір рет орындалуын қалаған кез келген уақытта пайдаланыңыз. Мысалы, файлға өлшемдердің белгіленген санын жазғыңыз келсе, файлға өлшемдерді жазатын кодты For Loop ішіне орналастырыңыз. Санау терминалына жалғанған сандық мән жазу әрекетінің қанша рет орындалатынын анықтайды. For Loop сіз орнатқан итерациялар санының ішкі диаграммасын орындағаннан кейін, For Loop тоқтайды.

Шарт орын алған кезде For циклін тоқтату үшін шартты терминалды қосыңыз. Шартты терминалды қоссаңыз, For Loop шарт орындалғанда немесе цикл сіз орнатқан итерациялар санын орындауды аяқтағанда тоқтайды.

For Loop әдепкі бойынша мәндерді бір итерациядан келесіге өткізбейді. Алдыңғы итерациядағы деректерге қол жеткізу және оны келесі итерацияда пайдалану үшін For Loop жүйесіне ауыстыру регистрлерін қосыңыз. Мысалы, келесі итерациядағы математикалық операцияның бастапқы мәні ретінде бір итерациядағы математикалық операцияның нәтижесін пайдаланыңыз келсе, ауыстыру регистрлерін пайдаланыңыз.

Массивті немесе жиынтық деректер түрін For циклінің кіріс туннеліне жалғаған кезде, For Loop әдепкі бойынша автоматты индекстеуді қосады. Автоматты индекстеу қосулы кезде, For Loop бүкіл массивті оқу мен өңдеудің немесе әрбір итерацияны жинаудың орнына, әрбір цикл итерациясында алаптағы немесе жинақтағы бір элементті оқиды және өңдейді. For Loop деректерін өңдеу әдісіне байланысты автоматты индекстеуді қосуға немесе өшіруге болады.

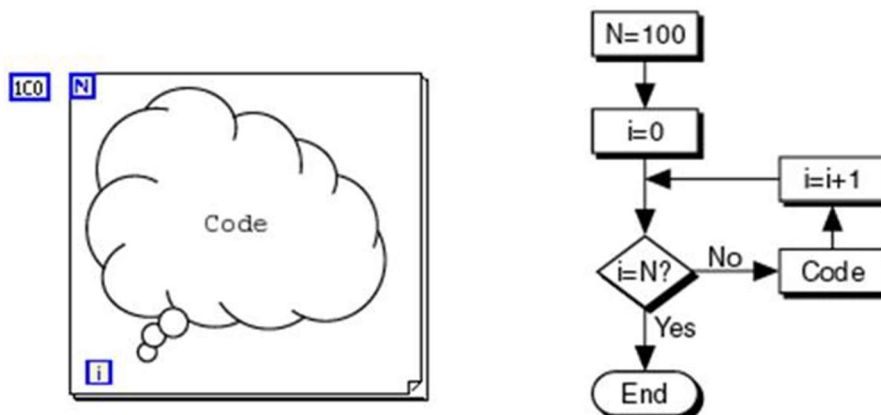
Егер For Loop жылдам орындалуы VI жұмысы үшін өте маңызды болса және VI жұмыс істейтін машинада бірнеше процессорлар болса, параллельді қайталауларды қосу қате немесе ескертуді тудырмаса, параллель For Loop итерацияларын қосыңыз. Параллельді итерацияларды қосқанда, процессорлар өнімділікті жақсарту үшін итерацияларды бір уақытта орындайды.

While циклі - берілген шарт орындалғанша LabVIEW кодының блогын қайталап орындау үшін пайдаланылатын құрылым. VI іске қосылғанда, While циклінің ішіндегі код орындалады, содан кейін терминал жағдайы бағаланады. While циклі тәжірибелі бағдарламашылар үшін таныс тұжырымдама болады, өйткені ол басқа компьютерлік тілдерде де жұмыс істейді.

For циклінен айырмашылығы, While циклінің орындалуы итерация санына тәуелді емес; осылайша, егер шарт ешқашан орын алмаса, While циклі шексіз орындалады.

LabVIEW ішіндегі for циклы

For [Loop](#) - бұл код блогын бірнеше рет орындау үшін пайдаланылатын құрылым. VI іске қосылғанда, итерация саны бағаланады, содан кейін код орындалады.



Сурет 1 - LabVIEW For Loop блок-схемасы

For Loop итерацияға негізделген шығуға қосымша кодтың орындалуын шартты түрде тоқтату үшін конфигурациялануы мүмкін. Бұл жағдайларда код санау терминалының параметріне жеткенше *немесе* шарт орындалғанша орындалады – қайсысы бірінші орын алса.

For Loops келесі жағдайларда қолданылуы керек:

- Кодты белгілі бір рет іске қосу керек;
- Файлға өлшемдердің белгіленген санын жазғыңыз келеді;
- Деректер нүктелерінің белгілі бір санын алғыңыз келеді;
- Шарт орындалғанша **НЕМЕСЕ** итерациялардың белгіленген саны, қайсысы бірінші келетініне дейін кодты іске қосқыңыз келеді.

LabVIEW ішінде For циклін құрастыру және конфигурациялау

For [Loop](#) - бұл код блогын бірнеше рет орындау үшін пайдаланылатын құрылым. VI іске қосылғанда, итерация саны бағаланады, содан кейін код орындалады. For Loops көптеген бағдарламалау тілдерінде кодты белгілі бір рет орындау қажет болғанда қолданылады.

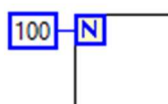
For циклін құрастыру

1. LabVIEW іске қосыңыз және жаңа VI ашыңыз. **Файл»Жаңа VI тармағын (File»New VI.)** таңдау арқылы бос VI ашуға болады.
2. Функциялар палитрасының уақытша нұсқасын көрсету үшін блок-схеманы тінтуірдің оң жақ түймешігімен басыңыз.
 - а. Қосымша: Бояғышты уақытша болмайтындай етіп бекіту үшін Функциялар палитрасының жоғарғы сол жақ бұрышындағы түймені басуға болады.
3. **Құрылымдар (Structures)** бөліміне өтіп, [For Loop](#) таңдаңыз.

- а. Циклды блок-схемаға орналастыру үшін тінтуірдің сол жақ батырмасын басып, цикл қажетті өлшемге жеткенше тінтуірді сүйреңіз.

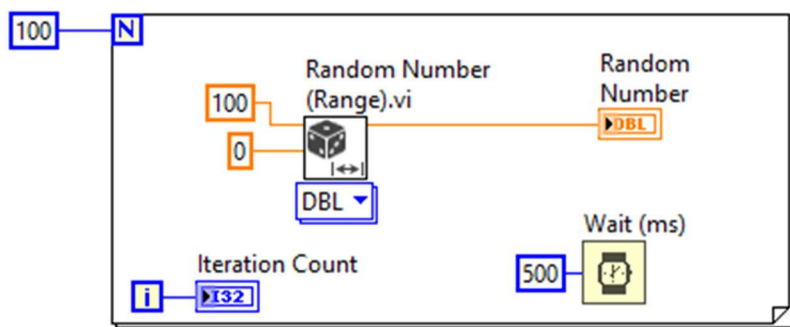


4. **Сандық терминалды тінтуірдің** оң жақ түймешігімен басып , сандық тұрақтыны көрсету үшін **Тұрақты құру пәрменін таңдаңыз.**
5. Санау терминалы ішкі диаграмманың қанша рет орындалатынын анықтайды. **Есептік терминал** константасына цикл орындалатын уақыт санын енгізіңіз .
- а. Мысалы, сандық тұрақтыға 100 енгізу арқылы for циклі тоқтау алдында 100 рет орындалады.



6. **Итерация терминалын** тінтуірдің оң жақ түймешігімен басып , **Көрсеткіш жасау пәрменін** таңдау арқылы цикл өткен итерациялар санын бақылаңыз

Төмендегі 2- суретте LabVIEW ішіндегі For Loop көрсетілген. Санау терминалы 100-ге орнатылған, сондықтан код циклден шықпас бұрын 100 рет орындалады. Циклдің әрбір итерациясында пайдаланушыға әрбір 500 мс сайын 0 мен 100 арасындағы кездейсоқ сан көрсетіледі. Итерация терминалы циклдің әрбір орындалуына қарай ұлғаяды.



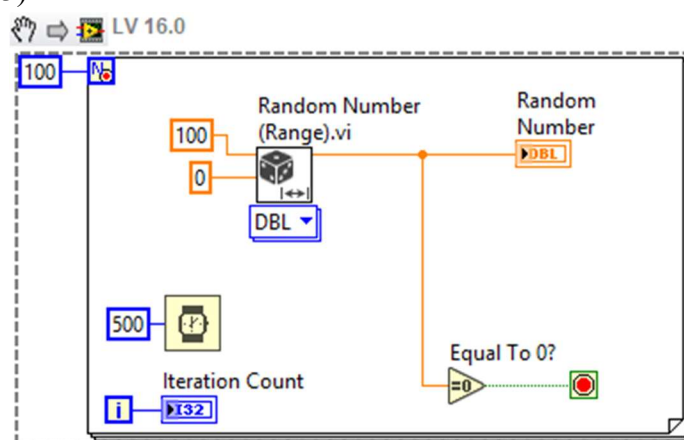
Сурет 2 - For Loop 100 кездейсоқ санды 0,5 сек интервалмен орындау мысалы

Шартты терминалды For цикліне қосу

Қажет болса, логикалық шарт орындалғанда немесе қате орын алған кезде тоқтау үшін for циклін конфигурациялау үшін шартты терминалды қосуға болады . Шартты терминалы бар for циклі шарт пайда болғанша немесе барлық итерациялар аяқталғанша, қайсысы бірінші орын алса, орындалады.

1. For циклінің жиегін тінтуірдің оң жақ түймешігімен басып, **Шартты терминалды** таңдаңыз .
 - а. Сіз санау терминалында қызыл глиф бар екенін және төменгі оң жақ бұрышта шартты терминал (яғни тоқтату глифі) қосылғанын байқайсыз.
2. For циклінің орындалуын тоқтатқыңыз келетін кодты қосыңыз.
 - а. Бұл пайдаланушы енгізуіне негізделуі мүмкін (мысалы, тоқтату түймесі, алдыңғы панельдегі мәннің өзгеруі және т.б.) немесе бағдарламалық талдауға негізделген (мысалы, кіріс мәнін өлшеу шекті мәнге жетеді, қате орын алады, уақыт өткен және т.б.)

Төмендегі үзінді жоғарыдағы *For Loop құрастыру* бөлімінде сипатталғандай кодты көрсетеді, бір ерекшелікпен – шартты терминал қосылған. Кездейсоқ жасалған сан нөлге тең болса *немесе* цикл 100 итерациядан өткен болса, бұл кодтың орындалуын тоқтатады (Сурет 3).



Сурет 3 - Шартты терминалды қосу мысалы

Параллельді цикл итерацияларының өнімділігін реттеу үшін теңшеу

Parallel For Loop итерацияларын қосу арқылы For Loop орындау жылдамдығын жақсартуға болады. Параллель итерацияларды қосу үшін келесі қадамдарды орындаңыз:

1. For Loop параметрін тінтуірдің оң жақ түймешігімен басып, Итерация параллелдігін конфигурациялау опциясын таңдаңыз . LabVIEW **цикл итерациясының параллелизмі үшін** диалогтық терезесін көрсетеді.
2. Цикл итерациясының параллелизмін қосу құсбелгісін қойыңыз .
3. Жасалған параллель цикл даналарының санын VI жүйесімен жұмыс істейтін кез келген компьютерде болуы күтілетін логикалық процессорлардың максималды санына орнатыңыз . Мысалы, ең көбі сегіз процессоры бар компьютерлерде қолдануға арналған қолданбаны әзірлеп жатсаңыз, санды сегізге қойыңыз.
4. ОК түймесін басыңыз. Параллель даналардың P терминалы конфигурацияланған For циклінің N санау терминалының астында пайда болады . Параллель итерацияларды қосу үшін мәнді параллель даналар терминалына жалғаудың

қажеті жоқ. Дегенмен, параллель For Loop өнімділігін өзгерту үшін терминалды пайдалануға болады .

Параллель итерацияларды қосқаннан кейін For Loop параллель итерациялармен жұмыс істей алатынын тексеріңіз .

Parallel For Loop итерацияларын қосқанда, P параллельді даналарды және бөлік өлшемі C терминалдарын пайдаланып өнімділікті бағдарламалы түрде реттеуге болады. Терминалдардың әдепкі конфигурациялары көп жағдайда оңтайлы өнімділікті жақсартады, сондықтан бағдарламалық конфигурация сирек қажет болады. Дегенмен, әдепкіден басқа конфигурацияны теңшеу үшін параллельді даналарды және бөлік өлшемі терминалдарын пайдалануға болады.

Parallel For Loop итерацияларын қосқаннан кейін параллель даналар терминалы For Loop ішінде пайда болады. Орындалатын параллель даналардың санын бағдарламалық конфигурациялау үшін сандық мәнді параллель даналар терминалына жалғаңыз. Компиляция уақытында LabVIEW Цикл Итерациясының Параллелдігі тілқатысу терезесінің Жасалған параллель цикл даналарының саны өрісінде және параллель даналар терминалының кірісіне жалғанған мәнде берілген мәннің ең азына тең параллель даналардың санын жасайды. Процессорлар өнімділікті жақсарту үшін параллельді даналарды бір уақытта орындайды.

Параллель даналар терминалының кірісін сымсыз қалдырсаңыз, LabVIEW компьютердегі логикалық процессорлардың санын автоматты түрде анықтайды және оны әдепкі параллель даналардың терминал мәні ретінде пайдаланады. Көп жағдайда оңтайлы өнімділік орындалатын параллель даналардың саны компьютердегі процессорлар санына тең болғанда орын алады, сондықтан әдетте параллель даналар терминалының кірісін сымсыз қалдыру керек.

For Loop ішіндегі код кез келген күту әрекетін орындаса, оңтайлы өнімділік орындалатын цикл даналарының саны артық жазылу деп аталатын логикалық процессорлар санынан көп болған кезде орын алады. Мысалы, егер параллельді For Loop данасы сыртқы жабдыктан деректерді алуды күтсе, артық жазылу процессорға біріншісін күтіп тұрғанда екінші параллель дананы орындауға мүмкіндік береді. Параллельді For Loop данасы басқа есептеуді қажет ететін кодпен бір уақытта орындалса, логикалық процессорлар санынан аз цикл даналарының санына төмен жазылу немесе орындау оңтайлы өнімділікке әкеледі. Мысалы, For Loop және subVI бір уақытта орындалса, төмен жазылу For цикліне арналған өңдеу ресурстарын шектейді және басқа әрекеттер үшін ресурстарды сақтайды.

LabVIEW бөлімдері итерацияларды цикл итерацияларынан тұратын бөліктерге айналдырады. Параллельді итерациялар қосылған кезде процессорлар орындау жылдамдығын жақсарту үшін бөліктерді бір уақытта орындайды. Әдепкі бойынша, LabVIEW бөліктерді өлшемі бойынша үлкеннен кішіге қарай жоспарлайды. Алдымен үлкенірек бөліктерді орындау жоспарлаудың үстеме шығындарын азайтады, ал кішірек бөліктерді соңғы орындау процессордың бос тұруын азайтады. Егер For Loop үлкенірек

бөліктерден бұрын кішірек бөліктерді орындайтын кесте сияқты әдепкіден басқа итерация кестесінен пайда көретін болса ғана бөлік өлшемін бағдарламалы түрде конфигурациялауыңыз керек.

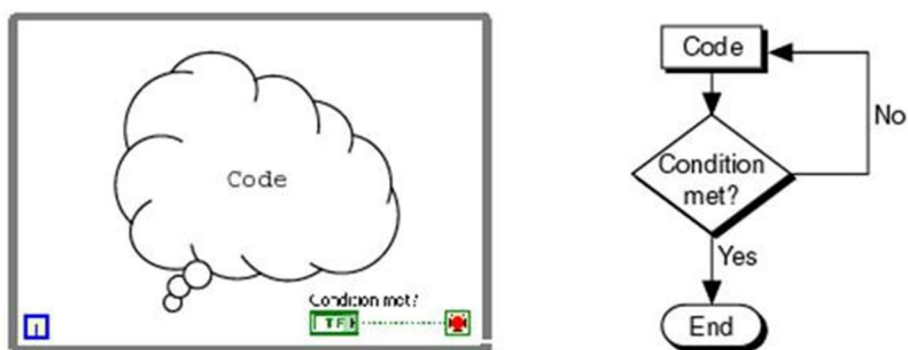
Параллель For Loop итерацияларын қосқаннан кейін бөлік өлшемі терминалын пайдаланып итерация кестесін бағдарламалы конфигурациялау үшін келесі қадамдарды орындаңыз:

1. For Loop параметрін тінтуірдің оң жақ түймешігімен басып, **Итерация параллелдігін конфигурациялау** опциясын таңдаңыз . LabVIEW **циклдік итерация үшін параллелизм диалогтық терезесін** көрсетеді .
2. **Итерацияны бөлу кестесі** бөлімінде **Бөлшек өлшемі (C)** терминалымен **бөлімді көрсету параметрін** таңдаңыз . Бөлшек өлшемі терминалы параллель даналар терминалының астында пайда болады.
3. Бөлшек өлшемі терминалына сандық мәнді немесе сандық мәндер массивін өткізіңіз.

Бөлшек өлшемі терминалына сандық мәнді қосқанда, мән әрбір бөлікке қосылатын итерациялар санын көрсетеді. Сандық мәндердің массивін қосу бөлік өлшемін дәлірек басқаруды қамтамасыз етеді. Жиымдағы әрбір мән бірінші бөлік 0 индексінен басталатын бөлікке қосылатын итерациялар санын көрсетеді.

LabVIEW ішіндегі while циклдері

While циклі - берілген шарт орындалғанша LabVIEW кодының блогын қайталап орындау үшін пайдаланылатын құрылым. VI іске қосылғанда, While циклінің ішіндегі код орындалады, содан кейін терминал жағдайы бағаланады. While циклі тәжірибелі бағдарламашылар үшін таныс тұжырымдама болады, өйткені ол басқа компьютерлік тілдерде де жұмыс істейді (Сурет 4).

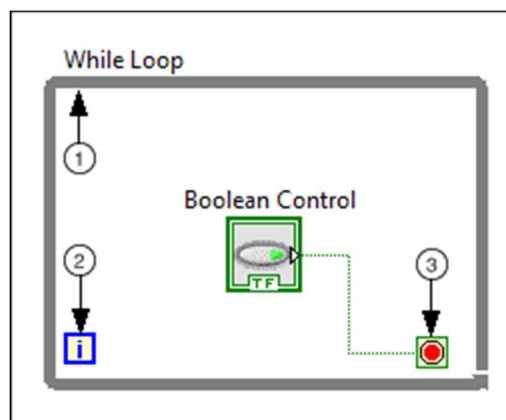


Сурет 4 - LabVIEW while циклінің блок-схемасы

For циклінен айырмашылығы, While циклінің орындалуы итерация санына тәуелді емес; осылайша, егер шарт ешқашан орын алмаса, While циклі шексіз орындалады

Арнайы шарт пайда болғанша, оның ішкі диаграммасындағы кодты қайталайды. While циклі әрқашан кем дегенде бір рет орындалады.

While циклінің компоненттері



Сурет 5 - While циклінің компоненттері

- ① — **Ішкі диаграмма** — While циклі әр итерацияда бір рет орындалатын кодты қамтиды.
- ② — **Итерация терминалы (i)** — ағымдағы цикл итерациясының санын қамтамасыз етеді. Бірінші итерация үшін цикл саны әрқашан нөлден басталады. Итерация саны 2 147 483 647 немесе $2^{31} - 1$ -ден асса, итерация терминалы барлық келесі итерациялар үшін 2 147 483 647 болып қалады. Егер сізге 2 147 483 647 итерация санын сақтау қажет болса, сіз үлкен бүтін диапазоны бар жылжу регистрлерін пайдалана аласыз.
- ③ — **Шартты терминал** — While циклін орындауды жалғастыруды анықтау үшін логикалық кіріс мәнін бағалайды. Циклдың TRUE немесе FALSE логикалық мәні үшін тоқтайтынын анықтау үшін циклдің жалғасу әрекетін конфигурациялаңыз. Сондай-ақ қате кластерін шартты терминалға жалғау арқылы циклдің қашан тоқтайтынын анықтауға болады.

While циклдері келесі жағдайларда қолданылуы керек:

- Сіз кодтың шексіз жұмыс істейтінін қалайсыз
- Шарт орындалғанша кодты іске қосқыңыз келеді
- Сіз кодты қашан тоқтату керектігін пайдаланушы бақылауын қалайсыз

While циклін құру

1. LabVIEW іске қосыңыз және жаңа VI ашыңыз. **Файл»Жаңа VI** таңдау арқылы бос VI ашуға болады
2. Функциялар палитрасының уақытша нұсқасын көрсету үшін блок-схеманы тінтуірдің оң жақ түймешігімен басыңыз.

- a. Қосымша: Бояғышты уақытша болмайтындай етіп бекіту үшін Функциялар палитрасының жоғарғы сол жақ бұрышындағы түймені басуға болады.

3. Құрылымдар бөліміне өтіп , While циклін таңдаңыз.

- a. Циклды блок-схемаға орналастыру үшін тінтуірдің сол жақ батырмасын басып, цикл қажетті өлшемге жеткенше тінтуірді сүйреңіз.



- 4. Алдыңғы панельге **Тоқтату** түймесін қосыңыз . Мұны **Басқару палитрасы» Логикалық »Тоқтату** тармағында таба аласыз .
- 5. Блок-сызбада Тоқтату түймешігінің белгішесін while цикліне апарыңыз.
- 6. Тоқтату түймесін шартты терминалға жалғаңыз, сонда сіз while циклінің орындалуын басқара аласыз.
 - a. Әдепкі бойынша, Тоқтату түймесі басылғанда, шын мән шартты терминалға беріледі, бұл while циклінің орындауды тоқтатуына әкеледі.
 - b. Ескерту: Тоқтату түймешігінің орнына уақытша циклдің орындалуын басқару үшін шартты терминалға кез келген логикалық деректерді қосуға болады.



- 7. Шартты терминалды конфигурациялаңыз. Шартты терминалды **«Шын болса тоқтату»** және **«Шын болса жалғастыру»** арасында ауыстыру үшін **шартты терминалды** тінтуірдің оң жақ түймешігімен басып , қажетті параметрді таңдаңыз.
 - a. Шартты терминал циклдің қашан тоқтайтынын анықтайды. Шартты терминалдың екі параметрі бар: **егер ақиқат болса, тоқтату** () және **шын болса, жалғастыру** ().
 - i. Шартты терминал Тоқта, егер True күйіне орнатылса және шын мәнінің логикалық мәні шартты терминалға жіберілсе, цикл орындауды тоқтатады. Бұл әдепкі параметр.
 - ii. Егер шын болса Жалғастыру күйіне орнатылғанда, уақытша цикл шын мәнінің логикалық мәні терминалға жіберілген жағдайда ғана іске қосылады.
- 8. **(Қосымша) Итерация терминалын** тінтуірдің оң жақ түймешігімен басып , **Көрсеткіш жасау пәрменін** таңдау арқылы цикл өткен итерациялар санын бақылаңыз .

While цикліне құрылымдық туннельдерді қосу

Уақыт циклі сияқты құрылымдарға деректерді беру және шығару үшін құрылым туннельдерін пайдаланыңыз. Уақыт цикліне деректерді жібергіңіз келсе, құрылымдық туннельдер жасауыңыз керек.

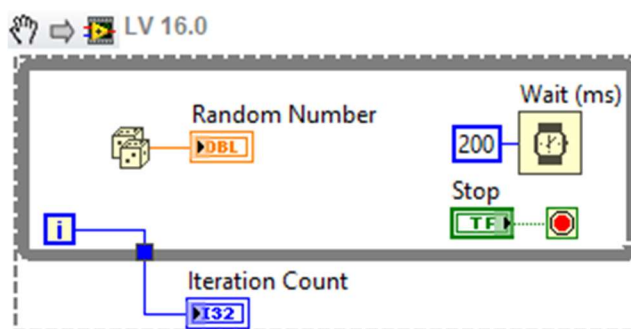
- *While* цикліне жіберілген деректер *тек бірінші итерацияда* жіберіледі
- *while* циклінен жіберілген деректер *тек соңғы итерациядан кейін* жіберіледі .
 - Алдыңғы панель деректерінің үздіксіз жаңартылуын қаласаңыз, олардың көрсеткіштері *while* циклінің ішінде орналасуы керек

Мысал: Итерациялардың аяқталғанын анықтаңыз

1. Уақыт циклін құру үшін қадамдарды аяқтаңыз (жоғарыда).
2. Алдыңғы панельде **сандық көрсеткіш** жасаңыз (**Басқару элементтері >> Сандық >> Сандық көрсеткіш**).
 - a. (Қосымша) Деректер түрлерін мәжбүрлеуді болдырмау үшін индикаторды тінтуірдің оң жақ түймешігімен басып, **Өкілдік >> I32 пәрменін** таңдау арқылы сандық көрсетілімді **I32** етіп өзгертіңіз (Сурет 6).
3. Блок-схемада итерация есептегішін сандық көрсеткіш белгішесіне жалғаңыз
 - a. Құрылым туннелі *while* циклінің шекарасында тұтас блок ретінде пайда болады. Блок туннельге қосылған деректер түрінің түсі болып табылады.
 - b. Мәліметтер цикл аяқталғаннан кейін циклден шығады.

Келесі блок-схемада итерация терминалы туннельге қосылған. Туннельдегі мән итерация индикаторына *while* циклі орындалуды аяқтамайынша берілмейді. Үзіндіні LabVIEW кодыңызға енгізіңіз және ескертіңіз:

- Кездейсоқ сан индикаторы *while* циклінің ішінде орналасқан. Бұл көрсеткішке арналған деректер әрбір иерархиямен жаңартылады.
- Итерация саны индикаторында код жұмыс істеп тұрған кезде нөлдік мән бар. Себебі, уақытша цикл орындалуды тоқтатпайынша, туннель индикаторға ақпаратты жібермейді.
- Тоқтату түймесін басқанда, итерация санының индикаторы енді жаңартылады.



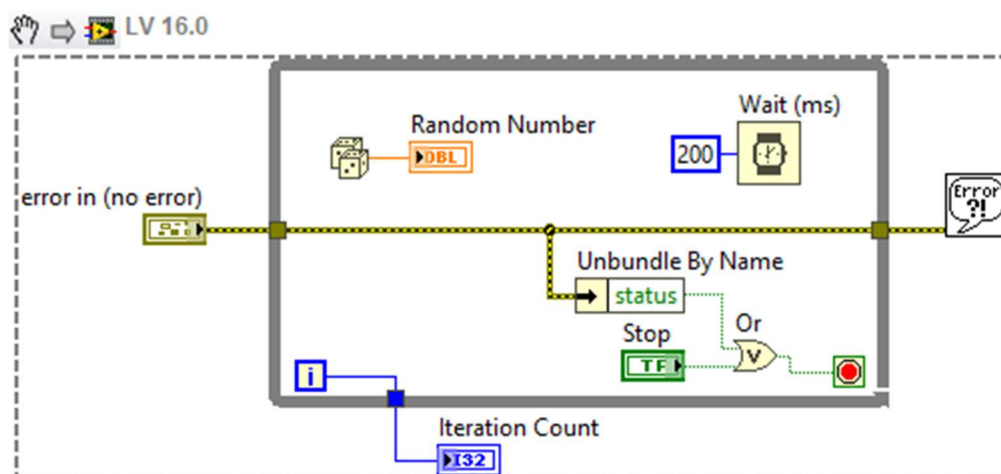
Сурет 6 - Кездейсоқ сандарды шығару

Тоқтату шарттарын орнату

Логикалық функцияларды пайдалану арқылы while циклінің шартты терминалына әсер ету үшін бірнеше шарттарды орындауға болады. Қате сымының күйін және Тоқтату түймешігін басқару элементін салыстыру үшін «немесе» функциясын пайдалануға болады, осылайша егер біреуі ШЫН болса, шартты терминал TRUE сигналын қабылдайды және while циклі тоқтайды.

1. Уақыт циклін құру үшін қадамдарды аяқтаңыз (жоғарыда).
2. Алдыңғы панельдегі **3D.ctf басқару элементінде қатені** орналастырыңыз (**Басқару элементтері палитрасы» Деректер контейнерлері» 3D.ctf қатесі**)
 - a. Блок диаграммасында 3D.ctf қатесі белгішесі while циклінің сыртында екеніне көз жеткізіңіз.
3. Тоқтату түймесі мен шартты терминал арасындағы қосылым сымын алыңыз.
4. Қарапайым қате өңдегішті while циклінің сыртына орналастырыңыз (**Функциялар палитрасы» Диалог және пайдаланушы интерфейсі» Қарапайым қате өңдегіші.vi**). Бұл функция шақырылған кезде кездесетін қателер туралы хабарлайды.
5. Басқарудағы қатені құрылым туннельдері арқылы while циклі арқылы Қарапайым қате өңдегішімен байланыстырыңыз.
 - a. Басқарудағы қатенің шығысын while циклінің шекарасына өткізіңіз. Туннель жасау үшін шекараны басыңыз.
 - b. Қате сымын туннельден while циклінің қарама-қарсы шекарасына дейін және Қарапайым қате өңдегішінің кірісіне жалғастырыңыз.
6. **Аты бойынша бөлу** функциясын while циклінің ішіне орналастырыңыз (**Функциялар палитрасы»Бағдарламалау»Кластер, класс және нұсқа»Аты бойынша бөлу**). Бұл функцияны қате деректер сымының логикалық бөлігін бөлу үшін пайдалануға болады.
 - a. Қате сымын Аты бойынша ажырату функциясына қосыңыз.
 - b. Unbundle функциясын тінтуірдің сол жақ түймешігімен басыңыз және қате деректерінің логикалық бөлігін көрсету үшін **күйді тандаңыз**.
7. while циклінің ішіне **немесе** функциясын орналастырыңыз (**Функциялар палитрасы» Бағдарламалау » Логикалық »Немесе**).
8. Деректер сымдарын байланыстырыңыз.
 - a. Немесе функциясына **Тоқтату** түймешігін басқару элементін және **Аты бойынша бөлу** күйін қосыңыз.
 - b. **Or** функциясының шығысын шартты терминалға қосыңыз.

Төмендегі үзінді уақытша циклдің орындалуын тоқтатады, егер Тоқтату түймесі басылса немесе уақытша цикл ішінде қате анықталса. (Сурет 7).

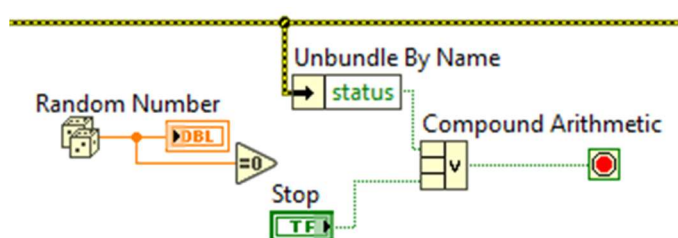


Сурет 7 - Тоқтату шарты бар цикл

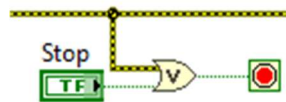
Тоқтатудың басқа жағдайлары

Жоғарыда сипатталған жағдай (пайдаланушы енгізуінде *немесе* қатеде тоқтату) шартты тоқтату енгізуін орнату үшін өте кең таралған жағдай болып табылады. Бірақ бұл тоқтау шарттарына күрделілік қосудың жалғыз жолы емес:

- Үш немесе одан да көп логикалық шарттарды салыстырғыңыз келсе, [Құрама арифметика](#) функциясын пайдаланыңыз. Бұл қате күйлерін, пайдаланушы енгізулерін және өлшеу деректерінің мәндерін бір уақытта бақылап, шек немесе күй орын алған жағдайда іске қосылған кодты тоқтатқыңыз келсе пайдалы болады.



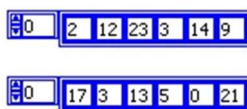
- Or, Құрама арифметика немесе басқа логикалық салыстыру функцияларын пайдаланып қате сымдарын салыстырған кезде қателер жинағын арифметикалық функцияға тікелей қосуға болады. Логикалық қате күйі автоматты түрде талданады.
 - Бұл LabVIEW бағдарламасының барлық нұсқалары үшін дұрыс болмауы мүмкін. *Тоқтату шарттарын орнату* бөлімінде сипатталған процесс LabVIEW бағдарламасының барлық нұсқаларында қолданылады және орындалатын дұрыс процесс.



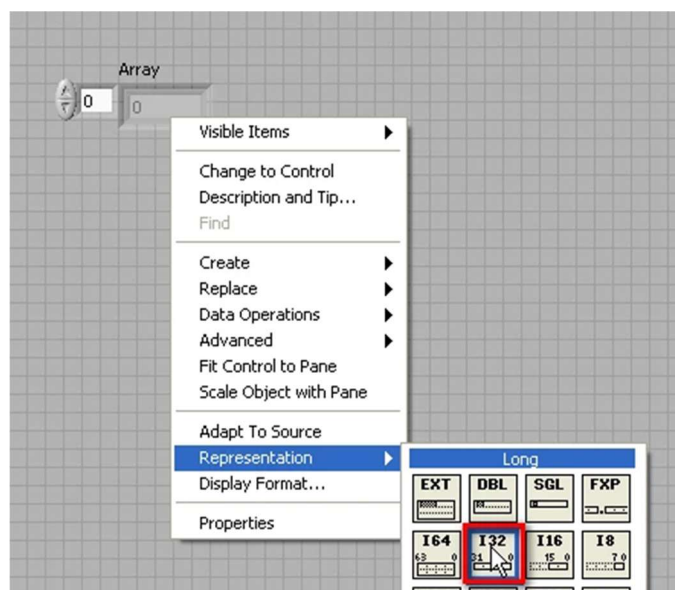
Жиым кірістері/шығыстары

Массивті for цикліне кіріс ретінде жалғасаңыз, LabVIEW автоматты индексстеу мүмкіндігін пайдаланып for циклінің санау терминалын массив өлшеміне автоматты түрде орнату опциясын қамтамасыз етеді. Жиымға жалғанған цикл туннелін тінтуірдің оң жақ түймешігімен басу және Индексстеуді қосу (Индексстеуді өшіру) пәрменін таңдау арқылы Автоматты индексстеу опциясын қосуға немесе өшіруге болады. Автоматты индексстеуді қоссаңыз, for циклінің әрбір итерациясына массивтің сәйкес элементі беріледі. Мәнді for циклінің шығысы ретінде қосқанда, Автоматты индексстеуді қосу массивді шығарады. Массив өлшемі бойынша for циклімен орындалатын итерациялар санына тең және for циклінің шығыс мәндерін қамтиды.

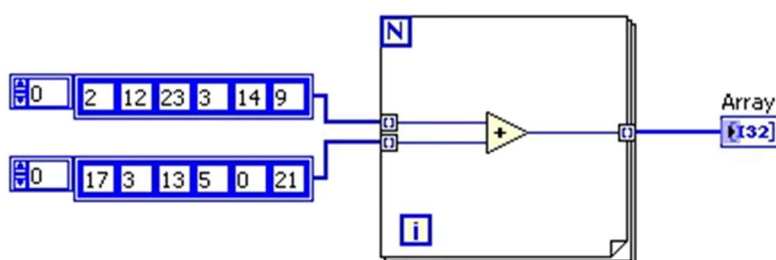
1. Жаңа VI жасаңыз. **Файл»Жаңа VI** тармағына өтіңіз .
2. Төменде көрсетілген массив константаларына ұқсас блок-схемада алты сандық элементі бар екі 1D жиым тұрақтысын жасаңыз және инициализациялаңыз.



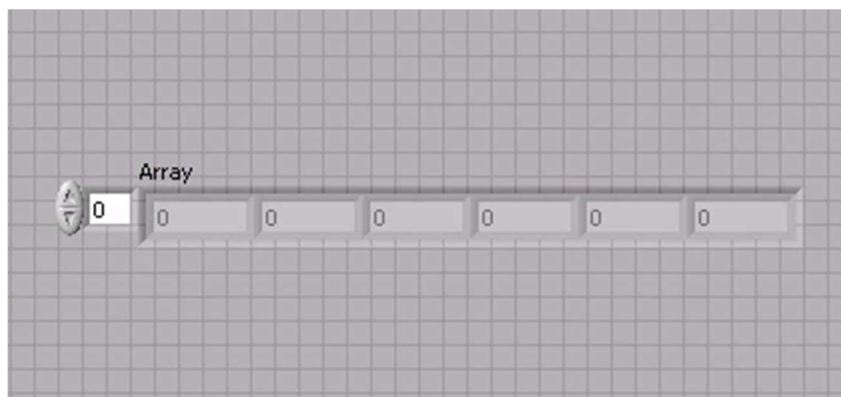
3. Алдыңғы панельде сандық көрсеткіштердің 1D массивін жасаңыз. Сан түрін 32 биттік бүтін санға өзгертіңіз. Жиымды тінтуірдің оң жақ түймешігімен басып, **Өкілдік»I32** таңдаңыз .



4. Блок-схемада for циклін жасаңыз және for циклінің ішіне қосу функциясын орналастырыңыз.
5. Жиым тұрақтыларының бірін for цикліне өткізіп, оны қосу функциясының x терминалына қосыңыз.
6. Басқа массив константасын for цикліне өткізіп, оны қосу функциясының y терминалына қосыңыз.
7. Қосу функциясының шығыс терминалын for циклінің сыртына өткізіп, оны сандық индикаторлар массивінің кіріс терминалына қосыңыз.
8. Соңғы блок диаграммаңыз және алдыңғы панель төменде көрсетілгендерге ұқсас болуы керек.

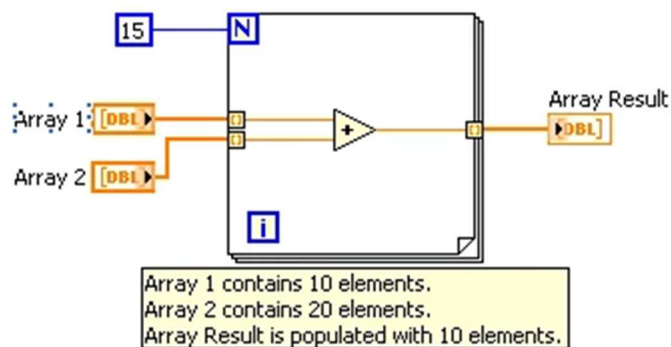


Блок-схема

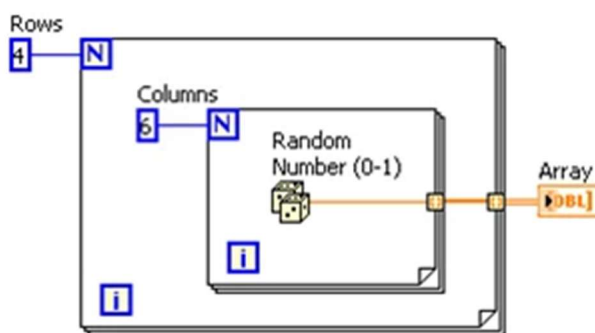


9. Алдыңғы панельге өтіп, VI іске қосыңыз. Сандық көрсеткіштер массивіндегі әрбір элемент екі жиым тұрақтысындағы сәйкес элементтердің қосындысымен толтырылатынын ескеріңіз.

Автоматты индекстеуді бірнеше цикл туннелінде қоссаңыз және for циклін санау терминалына сым қоссаңыз, итерация саны таңдаулардың кішісіне тең болатынын ескеріңіз. Мысалы, төмендегі суретте for циклін санау терминалы 15 итерацияны орындауға орнатылған, 1-массив 10 элементті, ал 2-жиым 20 элементті қамтиды. Төмендегі суреттегі VI орындасаңыз, for циклі 10 рет орындалады және массив нәтижесі 10 элементтен тұрады. Мұны көріңіз және оны өзіңіз көріңіз.



Төменде көрсетілгендей кірістірілген циклдар және Автоматты индекстеу арқылы 2D массивін жасауға болады. Сыртқы for циклі жол элементтерін жасайды, ал ішкі for циклі баған элементтерін жасайды.



Сурет 8 - Екіөлшемді массив шығару

Практикалық тапсырмалар

Тапсырма 1: Максималды мәнді табу (While цикл)

Шарты:

Пайдаланушы Stop батырмасын басқанға дейін While цикл кездейсоқ сандар (0–100 аралығында) генерациялайды.

Тапсырма:

- Генерацияланған сандарды Front Panel-де көрсет.
- Цикл тоқтаған кезде **барлық алынған сандар арасынан ең үлкен мәнді** шығаратын индикаторды жаса.

Кеңес: Shift Register және Max функциясын пайдалан.

Тапсырма 2: Массивті кері бұру (For цикл)

Шарты:

Пайдаланушы енгізген 1D массив (Array of I32) беріледі.

Тапсырма:

- For Loop арқылы массивтің элементтерін кері ретпен қайта жазатын VI құр.
- Шығару нәтижесі: бастапқы массивтің айналар бейнесі сияқты болуы керек.

Мысал: Кіріс: [5, 3, 7, 2]
Шығыс: [2, 7, 3, 5]

Тапсырма 3: Орташа мән және тоқтату шарты (While цикл)

Шарты:

While циклі пайдаланушы енгізген сандарды жинайды. Егер:

- 0 мәні енгізілсе → цикл тоқтайды.

Тапсырма:

- Цикл ішінде енгізілген сандардың **орташа мәнін** есепте.
- Тоқтағаннан кейін орташа мәнді Front Panel-де көрсет.

Кеңес: While цикл ішінде **сан қосындысы мен санауышты** сақтауға Shift Register қолдан.

Бақылау сұрақтары

1. **For және While** циклдерінің негізгі айырмашылықтарын жазыңыз. Олардың қайсысы қай жағдайда қолайлы екенін түсіндіріңіз.
2. **Shift Register** не үшін қолданылады? Оның циклдердегі маңызын мысал арқылы сипаттаңыз.
3. **While** циклінің шартты терминалы қалай жұмыс істейді? Оны қалай конфигурациялауға болады — түсіндіріңіз.
4. **For** циклінде автоматты индекстеудің маңызын түсіндіріңіз. Индекстеу қосулы және өшірулі кезде не болатынын салыстырыңыз.
5. **For** циклінде шартты тоқтату механизмін іске асыру жолын сипаттаңыз. Бұл үшін қандай терминалдар қажет және қалай конфигурацияланады?