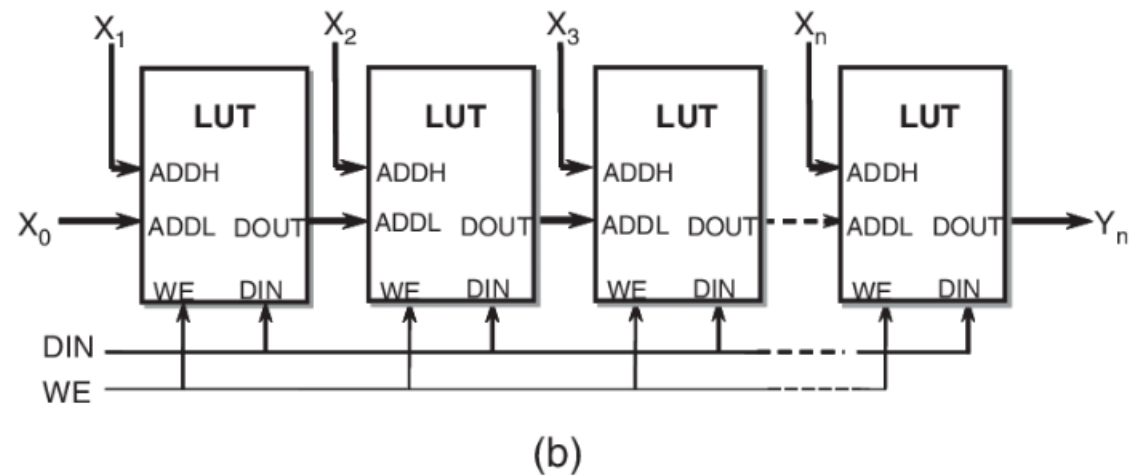
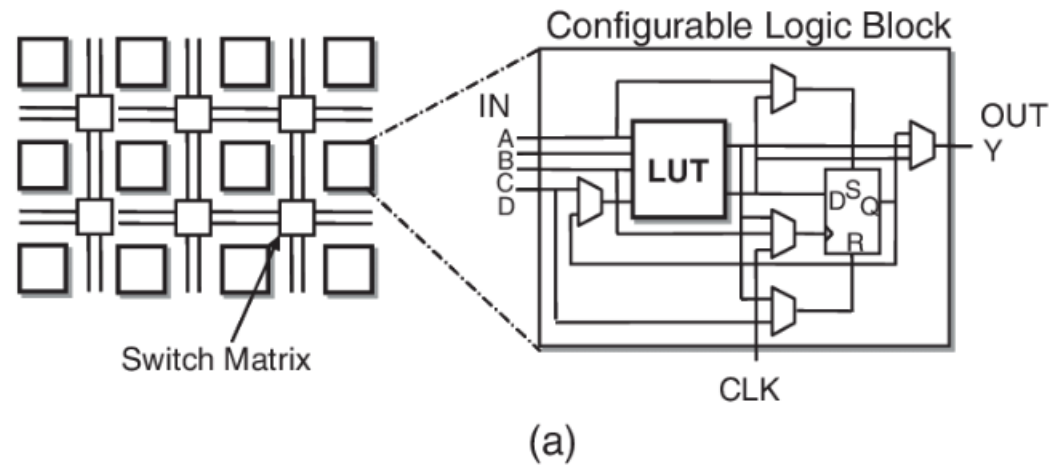


Триггерлер. Комбинациялық тізбекті талдау
және синтездеу. Синхронды және
асинхронды RS және JK триггерлері.

LUT

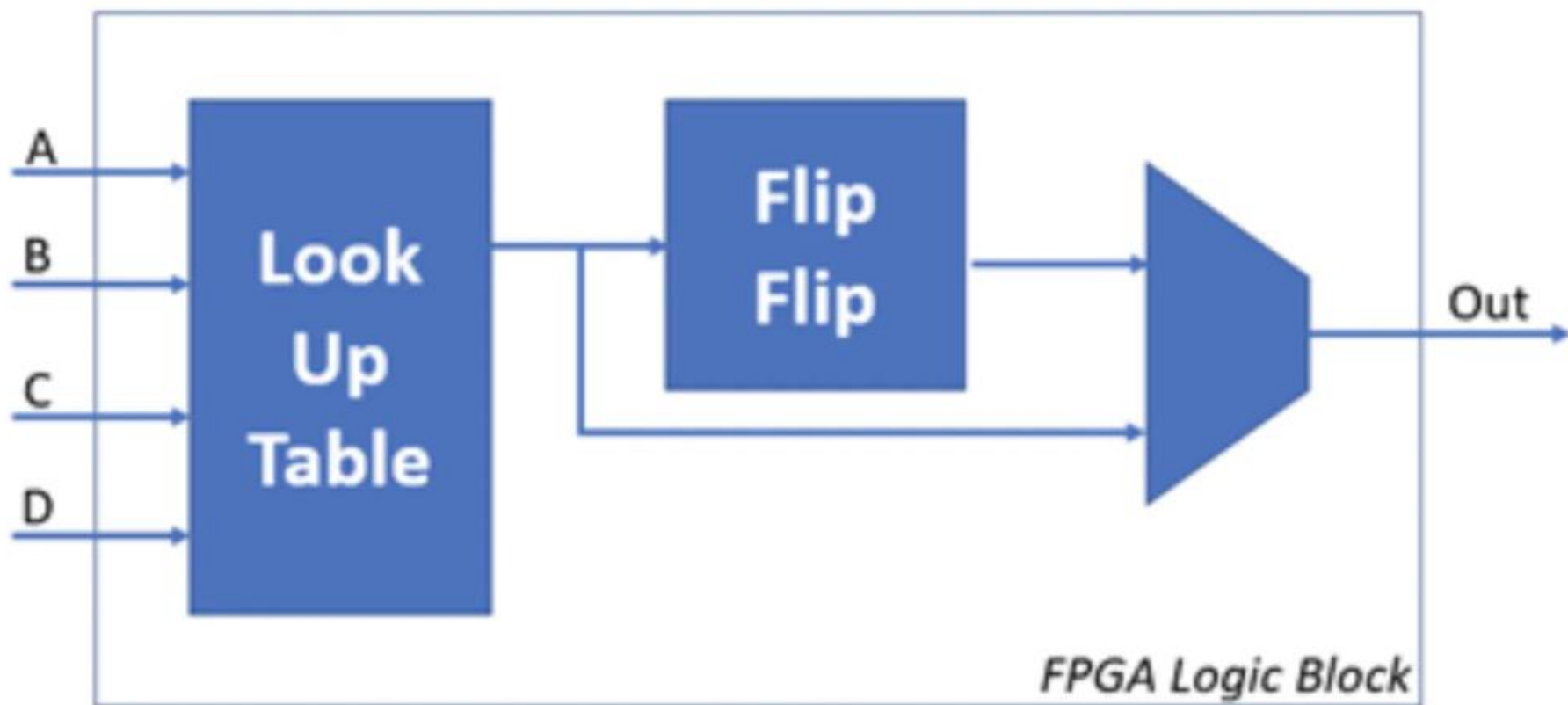
- Салыстырмалы түрде аз мөлшерде жедел жадты пайдаланатын LUTs CLB логикасының үлкен бөлігін жүзеге асыру үшін қолданылады. FPGA «жүйелік қақпалар» терминін пайдаланған кезде, ол берілген чиптегі NOR және NAND қақпаларының жалпы санына сілтеме жасайды. Дегенмен, барлық комбинаторлық логика, соның ішінде AND, OR, NAND, XOR және ұқсас операциялар, шын мәнінде LUT жадында ақиқат кестелері ретінде жүзеге асырылады. Шығыстардың кез келген мүмкін комбинациясы ақиқат кестесінде берілген.
- Іздеу кестелері (LUTs) логикалық алгебраны FPGA арқылы орындайтын құрал болып табылады. Іздеу кестесі сандық дизайнер бағдарламалай алатын ерекше функционалдық бірлік болып табылады. LUT шығыстарды кірістермен байланыстыру үшін бірдей ақиқат кестесі тұжырымдамасын пайдаланады. Virtex-7 FPGA, Kintex-7 FPGA, Virtex-6 FPGA, Spartan 6 FPGA және т.б. сияқты көптеген отбасылық серияларды қамтитын FPGA барлығы үш элементтен тұрады. мысалы Іздеу кестесі (LUT), Flip-Flop (FF), және сымдар.

Бағдарламаланатын логикалық архитектура



- FPGA шын мәнінде дискретті логикалық қақпаларды қамтымайды. Оның орнына, FPGAs LUT немесе іздеу кестелерін пайдаланады. Цифрлық дизайнер LUT-ті екі логикалық алгебралық теңдеуді шешуге бағдарламалайды. Болжам бойынша, іздеу кестесі барлық логикалық өрнек комбинацияларын орналастыру үшін бағдарламаланатын болуы керек. кез келген логикалық алгебра теңдеуін бір 3 кірісті LUT және үш кіріс сигналымен жасауға болады.
- Қолданылатын FPGA түріне байланысты LUT өлшемдері әртүрлі болуы мүмкін, бірақ олардың барлығы бірдей әрекет етеді. 3 кірісті LUT норма; бүгінде 4 кіріс және тіпті 5 кіріс LUT кең таралған. Күрделі өрнек жасау қажет болса, жай ғана қосымша іздеу кестелерін қосыңыз. FPGA-ның ең маңызды екі бөлігінің бірі - LUT. FPGA-да мыңдаған LUT бар.

FPGA ішіндегі LUT: Іздеу кестелері (LUTs) FPGA (Өріспен бағдарламаланатын қақпа массиві) дизайны үшін өте маңызды, өйткені олар күрделі сандық логикалық схемаларды жасауға мүмкіндік береді.



```
module lut4(input [3:0] in, output out);
reg [15:0] lut;

always @(*) begin
    case (in)
        4'b0000: lut = 16'b0000000000000001;
        4'b0001: lut = 16'b0000000000000010;
        4'b0010: lut = 16'b0000000000000100;
        4'b0011: lut = 16'b0000000000001000;
        4'b0100: lut = 16'b0000000000010000;
        4'b0101: lut = 16'b0000000000100000;
        4'b0110: lut = 16'b0000000001000000;
        4'b0111: lut = 16'b0000000010000000;
        4'b1000: lut = 16'b0000000100000000;
        4'b1001: lut = 16'b0000001000000000;
        4'b1010: lut = 16'b0000010000000000;
        4'b1011: lut = 16'b0000100000000000;
        4'b1100: lut = 16'b0001000000000000;
        4'b1101: lut = 16'b0010000000000000;
        4'b1110: lut = 16'b0100000000000000;
        4'b1111: lut = 16'b1000000000000000;
    endcase
end

assign out = lut[15];
endmodule
```

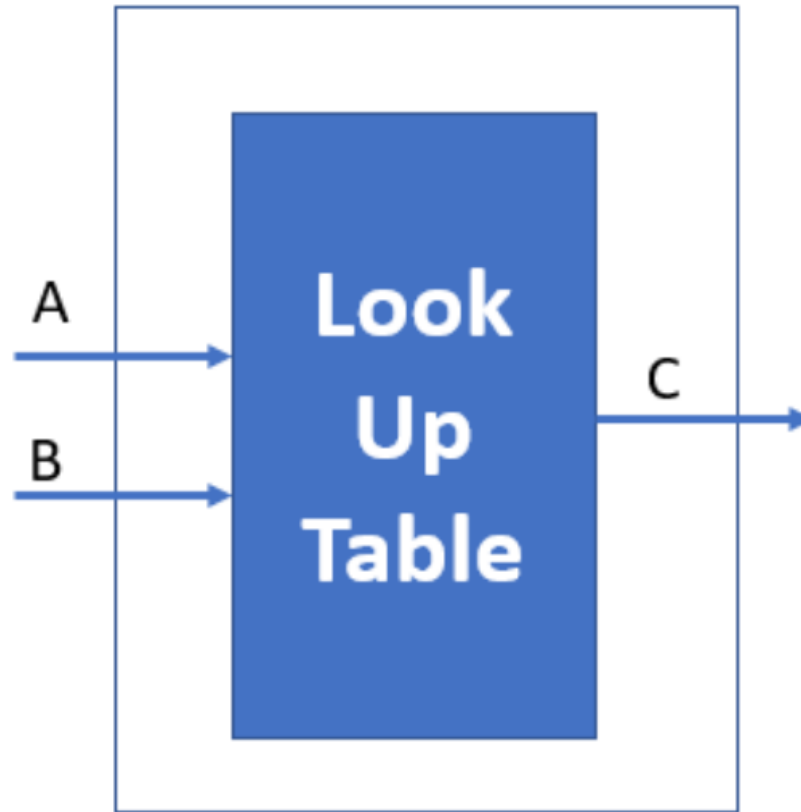
- Буль алгебрасынан белгілі элементтік негізді, мысалы, ЖӘНЕ-ЕМЕС немесе НЕМЕСЕ-ЕМЕС элементін пайдалана отырып, кез келген логикалық функцияны жүзеге асыруға болатыны белгілі. Дегенмен, күрделі құрылғыларды синтездеу кезінде тек бір ғана элемент түрін пайдалану техникалық тұрғыдан негізделмейді, көптеген элементтер сигналдың өту уақытын арттырады және сол арқылы өнімділікті төмендетеді. Сондықтан FPGA FPGA құрылымдарында қарапайым логикалық элемент ретінде күрделі құрылым пайдаланылады, ол бағдарламаланатын комбинациялық құрылғы мен D-флип-флоптың қосылуы болып табылады. 1-суретте мұндай элементтің жеңілдетілген құрылымы көрсетілген.

- Ұяшықта DATAA, DATAB және DATAC үш логикалық кірісі, CLK тактілік кірісі және LEOUT бір шығысы бар. Егер ұяшық тек комбинациялық құрылғы ретінде жұмыс істеуі қажет болса, онда шығыс мультиплексоры LUT элементінің шығысын бүкіл ұяшықтың шығысына ауыстырады, егер шығыс регистрлік болуы керек болса, онда LUT сигналы синхрондау арқылы бекітіледі сигнал D-флип-флопқа, оның шығысы мультиплексор арқылы LEOUT-ке қосылады. Мультиплексордың басқару кірісі (суретте көрсетілмеген) SRAM конфигурация жадысының сәйкес битіне қосылған.
- Егер триггер мен мультиплексордың жұмысы туралы ешкімде сұрақтар болмаса, диаграммада LUT ретінде көрсетілген элементпен бәрі біршама күрделірек. LUT аббревиатурасы «Іздеу кестесі» немесе жай іздеу кестесі дегенді білдіреді, оны сөзбе-сөз «іздеу кестесі» немесе «іздеу кестесі» деп аударуға болады. LUT - бұл кесте емес, LUT - бұл тікелей есептеу дайын шешімдер кестесі арқылы іздеумен ауыстырылатын функцияны жүзеге асыру әдісі; FPGA-ға қатысты бұл SRAM жады түріндегі кез келген логикалық функцияны жүзеге асыруға мүмкіндік береді, мұнда адрес аргумент, ал ұяшықтың мазмұны мән болып табылады. Осылайша, үш айнымалының логикалық қызметін сипаттау үшін (берілген мысалда үш айнымалы бар: DATAA, DATAB және DATAC) жадының 8 ұяшығы жеткілікті. Қажетті ақиқат кестесі маска (LUT-маска) ретінде сәйкес SRAM ұяшығында сақталады. Мультиплексорлардың көмегімен қажетті мән таңдалады. Мультиплексорлар кіріс портының сигналдарымен басқарылады, k айнымалылардың кез келген логикалық функциясын жүзеге асыратын k-кіріс LUT (k-LUT) құру үшін 2^k бит SRAM және $2^k - 1$ мультиплексорлар қажет. Төмендегі суретте үш кірісті LUT көрсетілген.

Іздеу кестесі, аты айтып тұрғандай, кірістерге негізделген шығысты жасайтын нақты кесте. Мұнда ЖӘНЕ қақпасының функциясын жүзеге асыратын іздеу кестесінің мысалы берілген.

Input A	Input B	Output C
0	0	0
0	1	0
1	0	0
1	1	1

Енді осы кестенің шағын жедел жадта сақталғанын елестетіп көріңіз. Мекенжай түйреуіштері А және В кірістері, ал деректер пиндері С кірісі болып табылады. Мекенжай түйреуіштеріңіз позицияларды өзгерткен сайын басқа мекенжай жазбасын көрсетеді және нәтижені «оқады», яғни 0 немесе 1 мәніне байланысты. кірістерде.



- Бірегей талаптарыңыз бен нұсқауларыңызға сәйкес іздеу кестесі - бұл сіздің жекелендірілген ақиқат кестеңіз, ол сіздің FPGA-ға сәйкес мәндермен толтырылады.
- Одан да жақсырақ, теңшелген іздеу кестесін (LUT) FPGA құрылғысы қосылған сайын жүктелетін жедел жадтың кішкене бөлігі ретінде елестетіңіз.
- VHDL немесе Verilog кодының негізінде ол жоспарланған нәтижелерге қол жеткізу үшін көрсетілген мәндерге сілтеме жасай отырып, құрылғыңыздың комбинаторлық логикасының әрекетін көрсетеді және басқарады.
- Нәтижесінде, іздеу кестесіне беретін кірістер негізінен ені бір бит жедел жад ұяшығы үшін мекенжай жолдары болып табылады.
- Сондықтан, екі кірісті логикалық қақпа төрт балама комбинаторлық мүмкіндіктерді генерациялай алады немесе орналастырады, бұл оны 4 x 1-биттік жедел жадқа сәйкес етеді.

LUT көмегімен логикалық функцияларды жүзеге асыру

- Чип әртүрлі комбинациялардағы логикалық функцияларды орындау үшін іздеу кестесін пайдаланады.
- Кез келген комбинаторлық логикалық операцияны жүзеге асыру үшін іздеу кестесін пайдалануға болады. FPGA-дағы іздеу кестелері уақытты немесе жоғары қақпалар санын қажет етпестен есептеу ауқымын қамтамасыз ететіндіктен, олар есептеу және пайдалану шығындарын айтарлықтай төмендетеді.
- SRAM FPGA процессорлары танымал болатын конфигурациялау мүмкіндіктерін ұсынады, бұл деректерді сақтау үшін пайдаланылған кездегі оның көптеген артықшылықтарының тағы бірі.
- Бұл чиптер қосылған сайын SRAM құрайтын жеке биттерді өзгертуге және өзгертуге болатынын білдіреді.

Іздеу кестелеріндегі ақаулар мәселесі

- FPGA чип конструкцияларындағы іздеу кестелері өте пайдалы болғанымен, жиі айтылатын бір кемшілік бар, ол ақаулық.
- Шығару не ретінде көрсетіліп жатқанда және ол немен синхрондалу керек болса, ақау пайда болады.
- Бұл Іздеу кестесінің SRAM биттерінің ауысуының және қайта реттелуінің нәтижесі.
- Осыған байланысты блок тиісті жауаптың қандай болуы керек екеніне сенімді емес және уақытша қате шығарылымды қамтамасыз етеді.
- FPGA чипін пайдалану арқылы сіз бұл мәселеге тап болмай қалуыңыз мүмкін, бірақ әрқашан бәріне дайын болу жақсы идея.

Іздеу кестелерінің артықшылықтары:

- Икемділік және конфигурациялау: Іздеу кестелері FPGA дизайнында икемділіктің жоғары деңгейін қамтамасыз етеді. Олар конструкторларға іздеу кестесінің ақиқат кестесінің жазбаларын бағдарламалау арқылы күрделі логикалық функцияларды анықтауға мүмкіндік береді, сандық схемалардың кең ауқымын жүзеге асыруға мүмкіндік береді.
- Ықшам көрініс: Іздеу кестелері логикалық функциялардың ықшам көрінісін ұсынады. Ақиқат кестесінің жазбаларын жадта сақтау арқылы Іздеу кестелері комбинаторлық логикалық желінің қажеттілігін жояды, нәтижесінде FPGA ресурстарын тиімді пайдаланады.

- Жылдамдық және өнімділік: іздеу кестелері жылдам және болжамды өнімділікті қамтамасыз етеді. FPGA ішкі маршруттауы мен өзара байланыстары логикалық функциялардың жылдам орындалуын қамтамасыз ете отырып, іздеу кестелерінен деректерге қол жеткізу және алу үшін оңтайландырылған.
- Уақыт бойынша мультиплекстелген әрекеттер: іздеу кестелерін жалғыз іздеу кестесінде бірнеше логикалық функцияларды немесе алгоритмдерді орындау үшін уақыт бойынша мультиплекстеуге болады. Бұл тәсіл дизайнерлерге ресурстарды барынша пайдалануды және жалпы FPGA дизайнын оңтайландыруға мүмкіндік береді.

- Орындаудың әмбебаптығы: Іздеу кестелерін күрделірек логикалық функцияларды жүзеге асыру үшін каскадты, біріктірілуі немесе тізбектелуі мүмкін. Бұл икемділік FPGA ішінде арифметикалық операцияларды, деректерді өңдеуді және күрделі алгоритмдерді жүзеге асыруға мүмкіндік береді.

Іздеу кестелерінің кемшіліктері:

- Шектеулі ажыратымдылық: Іздеу кестелерінің ақиқат кестесінде анықталған кірістер мен шығыстардың санымен анықталатын шектеулі ажыратымдылығы бар. Егер логикалық функция кірістердің немесе шығыстардың көп санын қажет етсе, ол бірнеше Іздеу кестелерін немесе балама дизайн әдістерін қажет етуі мүмкін.
- Ресурсты тұтыну: Күрделі логикалық функциялары бар конструкцияларда іздеу кестелерін пайдалану қол жетімді FPGA ресурстарының маңызды бөлігін тұтынуы мүмкін. Бұл ресурс шектеулеріне және жад блоктары немесе цифрлық сигналды өңдеу (DSP) блоктары сияқты басқа функционалдық элементтермен әлеуетті айырбастарға әкелуі мүмкін.

- Таралудың кешігуі: Іздеу кестелері жылдам орындауды қамтамасыз еткенімен, олар жадқа қол жеткізу және сәйкес шығысты алу үшін қажетті уақытқа байланысты таралу кешігуінің белгілі бір мөлшерін енгізеді. Сыни уақыт талаптары бар конструкцияларда бұл кідіріс қарастырылуы және оңтайландырылуы керек.
- Басқа ресурстармен айырбастау: іздеу кестелері жад блоктары және DSP блоктары сияқты FPGA ішіндегі ресурстар үшін бәсекелеседі. Дизайнерлер оңтайлы дизайнға қол жеткізу үшін іздеу кестелерін пайдалануды басқа функционалдық элементтердің талаптарымен мұқият теңестіруі керек.

- Шектеулі функцияны қайта пайдалану мүмкіндігі: Іздеу кестелері жобалау кезеңінде анықталған ақиқат кестесінің жазбаларына тән. Логикалық функцияны өзгерту Іздеу кестесін қайта бағдарламалауды немесе басқа конфигурацияны толығымен пайдалануды талап етуі мүмкін, бұл олардың белгілі бір сценарийлерде қайта пайдалану мүмкіндігін шектейді.

Триггерлер

- FPGA-ның ең маңызды екі бөлігінің бірі - іздеу кестесі (LUT), екінші орында флип-флоп келеді. Флип-флоптар бірнеше ерекше сорттарда келеді (JK, T және D). Негізгі флип-флоптардың ішінде JK флип-флоп ең бейімделгіш болып табылады. Сағатталған дәйекті логикалық схемаларда деректердің бір биті JK флип-флоп көмегімен сақталады. Ол SR флип-флоп сияқты дерлік орындайды. Жалғыз модификация S және R екеуі де 1 болатын анықталмайтын шартты жою болып табылады. JK флип-флопы төрт түрлі енгізу комбинациясын қабылдай алады, соның ішінде «логика 1», «логика 0», «өзгеріссіз» және «қосқыш, » қосымша уақытты енгізудің арқасында. Т SR флип-флопындағы аралық күйді болдырмау үшін («қосу») флип-флопты білдіреді.

- Аралық күйдің пайда болуын болдырмау үшін біз Trigger input Toggle кірісі ретінде белгілі флип-флопқа бір ғана кіріс беруіміз керек. Бірақ D Flip-Flop - ең танымал.
- Жоғарыдағы суретте S-белгіленген гаджеттің үстінде түйреуіш бар екенін көресіз. Set pin деп аталады. Құрылғының төменгі жағында R әрпі бар түйреуіш орналасқан. Қалпына келтіру пин - бұл солай аталады. Сондай-ақ құрылғының оң жағында «Qbar» деп белгіленген түйреуіш бар. Бұл түйреуіштегі мән әрқашан Q түйреуішіндегі мәнге қарама-қарсы полярлық болады. Flip-Flop-тың ең маңызды үш түйреуіштері:
- D: Флип-флопқа деректерді енгізу
- Q: Flip-Flop деректерінің шығысы (тіркелген)
- >: Флип-флопқа сағатты енгізу

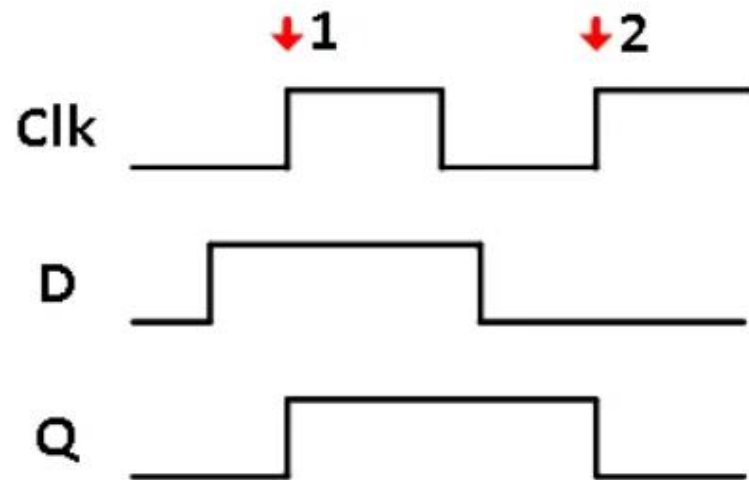
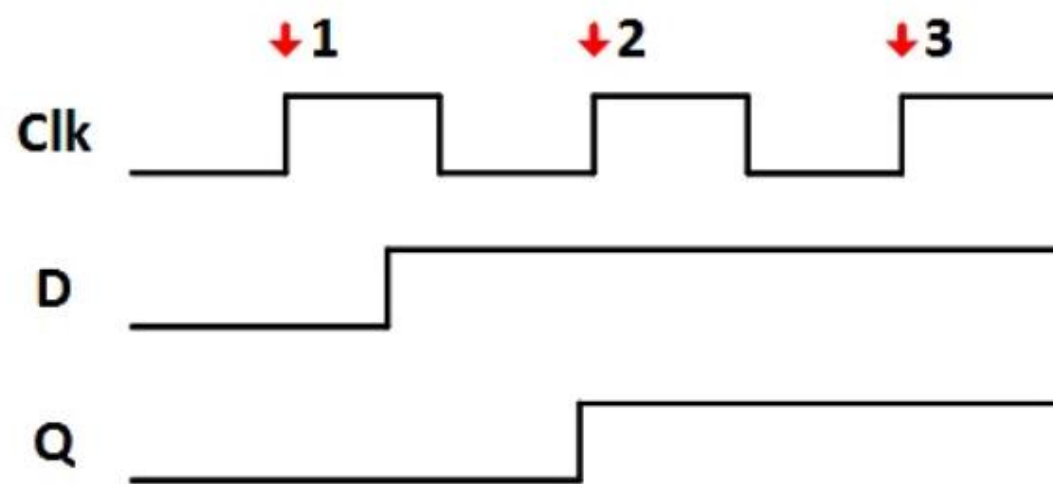
Сағат түрлері

- Барлық дерлік сандық схемалар сандық сағаттармен басқарылады. Жүйенің сағатын концептуализациялаудың бір тәсілі - берілістер тізбегі. Жүйедегі кез келген беріліс айналуы үшін негізгі беріліспен басқарылуы керек. Сандық логикада бұл негізінен сағаттың функциясы. Ол FPGA жұмыс істеуге мүмкіндік беретін төменнен жоғарыдан төменге кернеудің тұрақты ағынын ұсынады. Аналогия сонымен қатар қызықты, өйткені берілістер цифрлық сағаттағы шаршы толқынға ұқсайды.



- Жоғарыдағы толқын пішініндегі сағаттың көтерілу жиегіндегі қызыл көрсеткі сағат циклінің үш оқиғасын білдіреді. D кірісі сағаттың бірінші және екінші көтерілу жиектері арасында төменнен жоғарыға ауысады. Екінші тактілік циклдің көтерілу жиегі орын алғанда, Q шығысы D төменнен жоғарыға өзгергенін байқайды. Flip Flop берілген деректерді тексергенде, ол көтерілу шетінде болады. Q енді осы кезеңде D кірісімен бірдей мәнге ие. Осыған байланысты флип-флоптар жиі регистрлер деп аталады. Үшінші көтерілу жиегінде Q тағы бір рет D мәнін тексереді және оны тіркейді. Ештеңе өзгермегендіктен Q жоғары болып қалады.

- Екі кірісі және бір шығысы бар D Flip-Flop үшін толқын пішінін жоғарыдағы суретте көруге болады. Сағаттың көтерілу жиегі D Flip-Flop-қа әсер етеді, сондықтан ол орын алған кезде D кірісі Q шығысына ауысады. Ол тек шеттердің айналасында болады. Q бірінші тактілік цикл ішінде D 1-ге өзгергенін байқағаннан кейін 0-ден 1-ге ауысады. Q екінші сағат жиегінде D мәнін қайта тексергенде және оның тағы бір рет төмен екенін анықтағанда, ол қайтадан төмен болады.



- Егер FPGA тек LUT-тен тұруы мүмкін болса, есептік басқару мүмкін емес еді. Бұл FPGA бұрын не болғанын білмейтінін білдіреді. Өте ұзын тізбегі және қақпалар немесе қақпалар және т.б., кірістерге енгізілген барлық өзгерістер бірден талданып, шығысқа хабарланады. Дегенмен, FPGA функциялардың басым көпшілігін орындау үшін өткенді білуі керек. Ол осы жолмен есептегіштерді, күй машиналарын және нысан күйін бақылай алады. Флип-флоптар бұған мүмкіндік береді. Аппараттық құралды сипаттау тілімен (HDL) неғұрлым көп жұмыс жасасаңыз, соғұрлым FPGA дизайнындағы флип-флоптардың маңыздылығын түсінесіз.

Verilog және SystemVerilog жүйелерінде SR флип-флоптарын жобалау

```
module SRFF (  
    input  logic S, R, clk, rst,  
    output logic Q, Qn  
);  
    always_ff @(posedge clk, posedge rst) begin  
        if (rst) begin  
            Q  <= 0;  
            Qn <= 1;  
        end else if (S) begin  
            Q  <= 1;  
            Qn <= 0;  
        end else if (R) begin  
            Q  <= 0;  
            Qn <= 1;  
        end  
    end  
end  
endmodule
```