Жүйелерді FPGA негізінде жобалау

Дәріс 10: FPGA негізінде цифрлық жүйені практикалық іске асыру

ДӘРІСКЕР: PhD, доцент м.а.
Сқабылов Әлішер Әлиұлы

Дәрістің мақсаты:

FPGA негізінде цифрлық жүйені жобалау және оны нақты аппараттық құрылғыда іске асырудың толық процесін қарастыру, алдыңғы дәрістерде меңгерілген цифрлық логика, HDL, FSM, тактілік сигналдар, интерфейстер және жад туралы білімдерді біртұтас практикалық жобаға біріктіру.

Практикалық FPGA жобасына қойылатын талаптар

FPGA жобасын әзірлеу жүйенің атқаратын қызметін және жұмыс шарттарын анықтаудан басталады. Алдымен кіріс сигналдары, шығыс нәтижелері, басқару командалары және аппараттық шектеулер белгіленеді.

Мысал ретінде UART арқылы басқарылатын PWM жүйесін қарастырайық. Компьютер UART интерфейсі арқылы FPGA-ға команда жібереді. FPGA команданы қабылдап, басқару мәнін сақтайды және сол мәнге сәйкес PWM сигналын қалыптастырады. Бұл сигнал LED жарықтығын басқару үшін қолданылуы мүмкін.

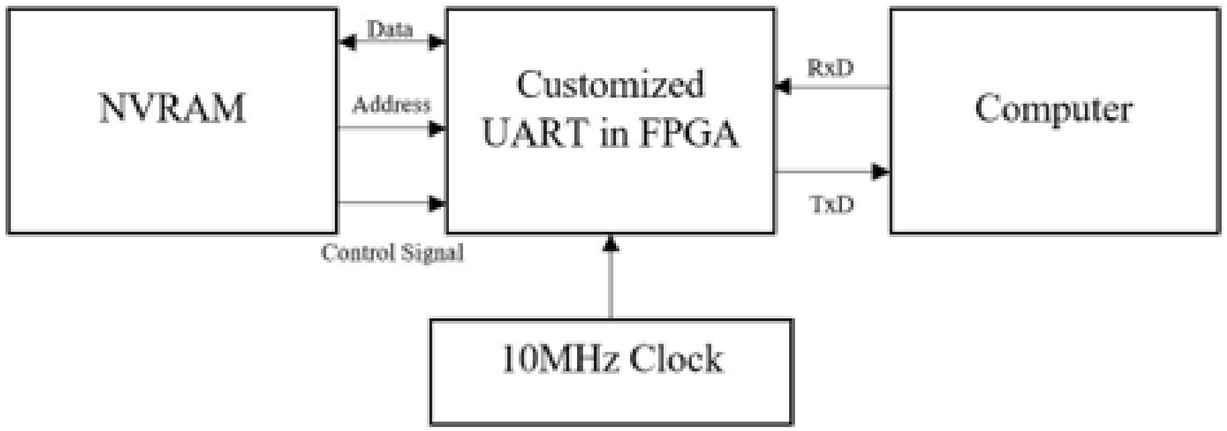
Жобалау талаптары екіге бөлінеді: функционалдық талаптар жүйенің не істеу керектігін анықтайды, ал аппараттық талаптар оның нақты FPGA тақтасында қалай іске асырылатынын белгілейді. Мысалы, UART және PWM сигналдары тиісті физикалық pin-дерге қосылып, жүйе тақтаның тактілік сигналымен жұмыс істеуі керек.

Жобаның аппараттық архитектурасы

Аппараттық архитектура жүйенің функционалдық блоктарын және олардың арасындағы деректер ағынын көрсетеді. UART арқылы басқарылатын PWM жүйесінің негізгі құрылымы:

UART Receiver → Data Processing → Control Register → PWM Generator → Output

UART Receiver компьютерден келетін деректерді қабылдайды. Data Processing модулі алынған ақпаратты талдап, қажетті басқару мәнін анықтайды. Control Register осы мәнді сақтайды, ал PWM Generator оны пайдаланып импульстік сигнал қалыптастырады.



Clock және *Reset* сигналдары синхронды модульдердің жұмысын басқарады. Әр блоктың міндетін нақты бөлу жүйені түсінуді, жеке модульдерді тексеруді және жобаны кейін кеңейтуді жеңілдетеді.

HDL модульдері және Top Module

Жүйенің әр функционалдық бөлігі HDL тілінде жеке модуль ретінде сипатталады. Бұл тәсіл күрделі жобаны бірнеше басқарылатын бөлікке бөлуге мүмкіндік береді.
Негізгі модульдер: UART Receiver, Data Processing, Control Register және PWM Generator. Олардың әрқайсысының кірістері, шығыстары және нақты қызметі анықталады.

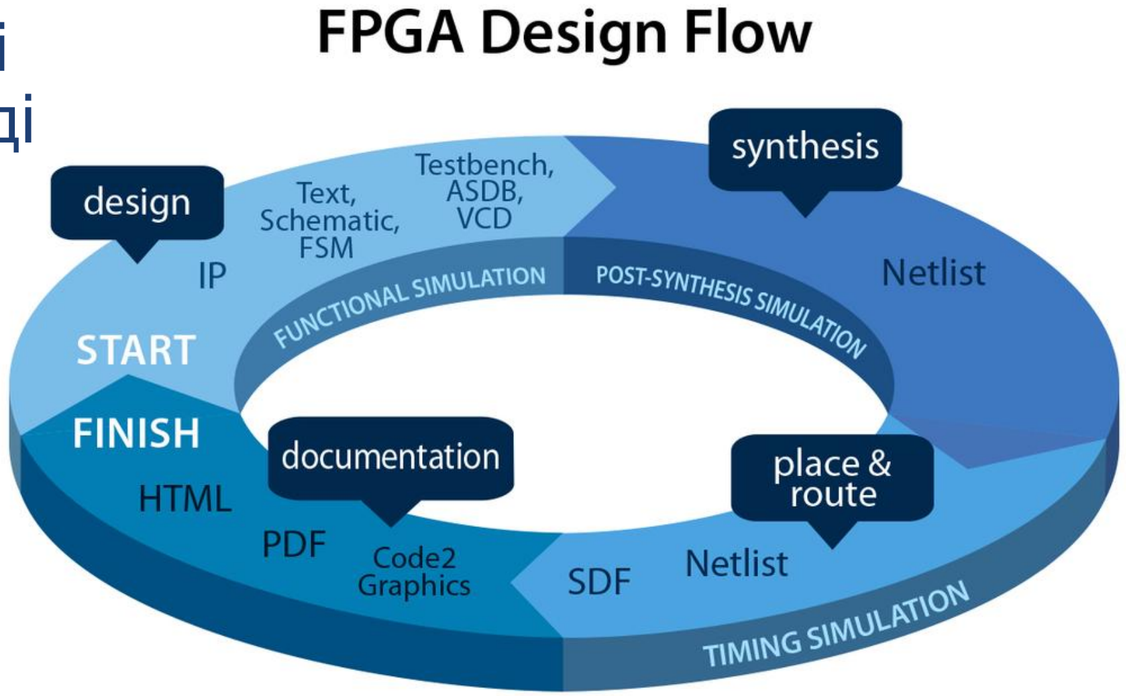
Top Module — барлық модульдерді бір жүйеге біріктіретін жоғарғы деңгейдегі модуль. Ол сыртқы clock, reset және UART RX сигналдарын қабылдап, ішкі модульдер арасындағы байланысты ұйымдастырады. Соңғы PWM сигналын FPGA-ның қажетті шығысына береді.

Модульдік жобалаудың *артықшылығы* — әр блокты жеке тексеруге және өзгертуге болады. Мысалы, UART модулі дұрыс жұмыс істесе, бірақ PWM шығысы күтілген нәтижені бермесе, басқару регистрі мен PWM Generator бөлек тексеріледі.

Simulation және функционалдық тексеру

Simulation — жобаны нақты FPGA құрылғысына жүктемей тұрып, оның логикалық жұмысын модельдеу арқылы тексеру процесі. Тексеру кезінде кіріс сигналдары беріледі, ал шығыс және ішкі сигналдардың өзгеруі бақыланады.

Testbench тексерілетін модульге қажетті сигналдарды беріп, күтілетін нәтижелерді салыстыруға мүмкіндік береді. UART-PWM жүйесін тексеру кезінде алдымен Reset беріледі. Одан кейін UART арқылы басқару командалары жіберіліп, басқару мәнінің дұрыс өзгергені және PWM сигналының соған сәйкес қалыптасқаны тексеріледі.



Synthesis және FPGA ресурстарын бағалау

Synthesis — HDL сипаттамасын FPGA ішінде іске асырылатын логикалық құрылымға түрлендіру процесі. Синтез нәтижесі жобаның қандай аппараттық ресурстарды пайдаланатынын көрсетеді.

Негізгі бағаланатын ресурстар:

- LUT — комбинациялық логикалық функциялар;
- Flip-Flop — деректер мен күйді сақтау;
- RAM — деректерді сақтау;
- DSP — арнайы арифметикалық операциялар.

UART-PWM сияқты қарапайым жобада LUT пен Flip-Flop негізгі ресурстар болуы мүмкін. Қосымша жад немесе күрделі есептеу блоктары енгізілсе, басқа ресурстардың қажеттілігі де артады.
Ресурстарды пайдалану есебі жобаның FPGA сыйымдылығына сәйкестігін бағалауға мүмкіндік береді. Синтез кезінде табылған ескертулер мен қателер де талданып, қажет болса HDL сипаттамасы оңтайландырылады.

Implementation, Pin Assignment және Bitstream

Synthesis аяқталғаннан кейін Implementation кезеңі орындалады. Бұл кезде синтезделген логикалық құрылым FPGA-ның нақты физикалық ресурстарына орналастырылады және олардың арасындағы routing байланыстары анықталады.

Pin Assignment HDL сигналдарын FPGA микросхемасының нақты физикалық pin-дерімен сәйкестендіреді. Мысалы, UART RX сигналы тақтадағы тиісті кіріске, ал PWM шығысы LED немесе басқа жүктемені басқаруға арналған pin-ге тағайындалады.

Constraint файлдары тактілік сигнал, енгізу/шығару стандарттары және басқа физикалық не уақыттық талаптарды сипаттауға көмектеседі.

Implementation сәтті аяқталғаннан кейін FPGA-ны конфигурациялауға арналған Bitstream файлы жасалады. Бұл файл құрылғыға жүктелетін дайын конфигурациялық ақпаратты қамтиды.

FPGA-ға жүктеу, аппараттық тестілеу және debugging

Bitstream файлы FPGA тақтасына жүктелгеннен кейін жүйенің нақты аппараттық жұмысы тексеріледі. UART-PWM мысалында компьютерден команда жіберіліп, PWM сигналының дұрыс өзгеруі бақыланады. Егер PWM LED-ті басқарса, LED жарықтығы басқару мәніне сәйкес өзгеруі мүмкін. Тестілеу кезінде бастапқы күй, ең кіші және ең үлкен басқару мәндері, бірнеше командаңың қатар берілуі және жарамсыз деректер тексеріледі.
Егер модельдеу дұрыс нәтиже беріп, бірақ нақты FPGA дұрыс жұмыс істемесе, pin assignment, clock, reset, интерфейс параметрлері мен сыртқы қосылыстар қарастырылады.

Debugging процесінде сигналдарды кезең-кезеңімен тексеру тиімді:

UART RX → UART Data → Data Valid → Control Value → PWM

Осылайша ақаудың қай модульде пайда болғанын анықтап, жобаны түзетуге болады. FPGA жобалау циклі тестілеу нәтижесіне қарай бірнеше рет қайталануы мүмкін.

Білімді бекітуге арналған сұрақтар:

1. FPGA жобасын әзірлеу қандай кезеңнен басталады?
2. Функционалдық және аппараттық талаптардың айырмашылығы қандай?
3. UART арқылы басқарылатын PWM жүйесінің негізгі блоктарын атаңыз.
4. Top Module қандай қызмет атқарады?
5. Simulation мен Testbench не үшін қолданылады?
6. Synthesis процесінің негізгі нәтижесі қандай?
7. Resource Utilization дегеніміз не?
8. Implementation кезінде қандай жұмыстар орындалады?
9. Pin Assignment және Constraint файлдарының маңызы қандай?
10. Bitstream файлы не үшін қажет?
11. Simulation дұрыс, бірақ нақты FPGA дұрыс жұмыс істемесе, қандай себептер болуы мүмкін?
12. Debugging кезінде сигналдарды кезең-кезеңімен тексерудің қандай артықшылығы бар?

Әдебиеттер тізімі:

1. Brown S., Vranesic Z. Fundamentals of Digital Logic with Verilog Design. McGraw-Hill.
2. Harris D., Harris S. Digital Design and Computer Architecture. Morgan Kaufmann.
3. Chu P. P. FPGA Prototyping by Verilog Examples. Wiley.
4. Wakerly J. F. Digital Design: Principles and Practices. Pearson.
5. Roth C. H., Kinney L. L. Fundamentals of Logic Design. Cengage Learning.
6. AMD. Vivado Design Suite User Guide: Synthesis. AMD Documentation.
7. AMD. Vivado Design Suite User Guide: Implementation. AMD Documentation.
8. AMD. Vivado Design Suite User Guide: Programming and Debugging. AMD Documentation.
9. Intel. Quartus Prime Pro Edition User Guide: Design Compilation. Intel Documentation.
10. Intel. Quartus Prime Pro Edition User Guide: Design Constraints. Intel Documentation.