Жүйелерді FPGA негізінде жобалау

Дәріс 9: FPGA жобасын әзірлеу, синтездеу және верификациялау процесі

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

Дәрістің мақсаты:

FPGA негізіндегі цифрлық жобаның толық әзірлеу процесін жүйелі түрде түсіндіру, RTL жобасынан бастап synthesis, implementation, timing analysis, конфигурациялық файлды қалыптастыру және hardware verification кезеңдеріне дейінгі өзара байланысты көрсету.

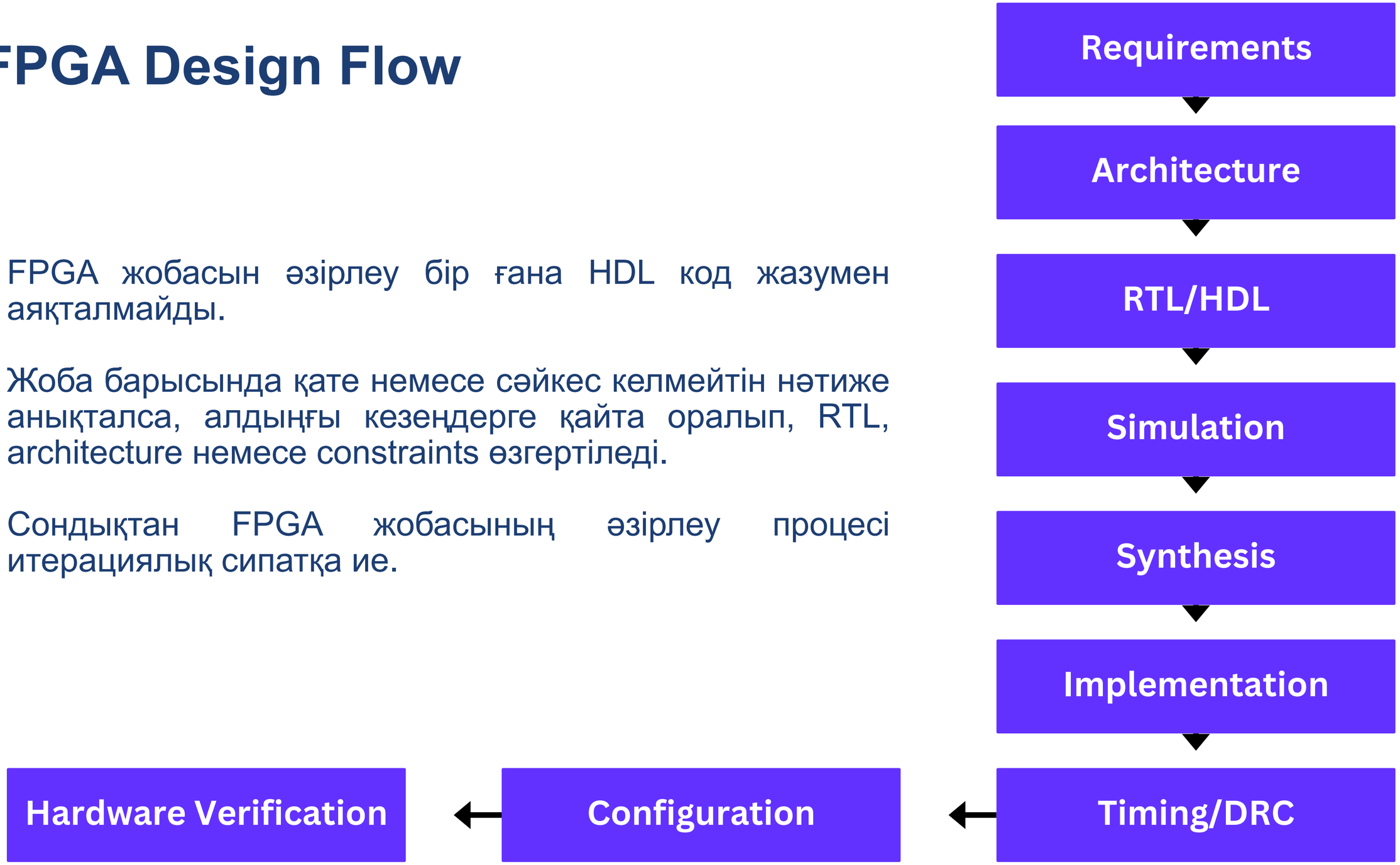
Міндеттері: Жобаның талаптары мен архитектурасын анықтау, RTL құрылымын ұйымдастыру, simulation және testbench арқылы тексеру, synthesis және implementation нәтижелерін талдау, timing constraints пен DRC қолдану, FPGA конфигурациясын орындау және нақты құрылғыдағы debug процесін түсіндіру.

FPGA Design Flow

FPGA жобасын әзірлеу бір ғана HDL код жазумен аяқталмайды.

Жоба барысында қате немесе сәйкес келмейтін нәтиже анықталса, алдыңғы кезеңдерге қайта оралып, RTL, architecture немесе constraints өзгертіледі.

Сондықтан FPGA жобасының әзірлеу процесі итерациялық сипатқа ие.



Жобаның талаптарын анықтау және архитектура

Жобалау алдымен жүйенің нақты не істеуі керегін анықтаудан басталады.
Негізгі талаптарға мыналар жатады:

Functional requirements — жүйенің орындайтын функциялары.

Performance — қажетті жұмыс жиілігі және өнімділік.

Latency — деректерді өңдеу кідірісі.

Interface — сыртқы құрылғылармен байланыс тәсілдері.

Resource — LUT, FF, BRAM, DSP және басқа ресурстардың шектеулері.

Reliability — reset, synchronization және CDC талаптары.

Талаптар анықталғаннан кейін жүйенің архитектурасы құрылады: негізгі модульдер, олардың сигналдары, data flow және clock domain-дер белгіленеді.

RTL жобалау

RTL (Register-Transfer Level) – деректердің регистрлер арасында қалай өңделіп, тасымалданатынын сипаттайтын жобалау деңгейі.
FPGA жобасы әдетте:

Top-Level Module → Internal Modules
құрылымынан тұрады.

Top-level module FPGA pin-дері және сыртқы интерфейстермен байланысады.

Ішкі модульдер:
есептеу + басқару + сақтау + интерфейс
функцияларын орындай алады.

Жобада module hierarchy, compilation order және constraints файлдарын дұрыс ұйымдастыру маңызды.

Simulation және Testbench

Simulation FPGA-ға нақты жүктемей тұрып жобаның логикалық мінез-құлқын тексеруге мүмкіндік береді.

Testbench арқылы:
Clock
Reset
Input Data
Control Signals
қалыптастырылады және output мәндері күтілетін нәтижемен салыстырылады.

Күрделі жобаларда:
Boundary Cases → Invalid Inputs → Reset Sequence → Different Operating Modes
сияқты жағдайлар да тексеріледі.

Simulation функционалдық қателерді ерте анықтауға мүмкіндік береді, бірақ нақты physical timing мәселелерін толық алмастырмайды.

Synthesis

Synthesis – RTL/HDL сипаттамасын мақсатты FPGA-ның логикалық ресурстарына сәйкестендірілген netlist-ке түрлендіру процесі.

Жалпы:
RTL/HDL → Synthesis → Netlist

Synthesis кезінде:
логикалық оптимизация орындалуы мүмкін;
артық логика жойылуы мүмкін;
логикалық функциялар FPGA ресурстарына сәйкестендіріледі;
inferred structures анықталады.

Негізгі нәтижелердің бірі – **resource utilization** туралы ақпарат:
LUT + Flip-Flop + BRAM + DSP + I/O

Implementation: Placement және Routing

Synthesis-тен кейін netlist FPGA ішінде физикалық түрде жүзеге асырылады.

Placement
Логикалық элементтер, registers, memory және басқа ресурстар FPGA ішіндегі нақты физикалық орындарға орналастырылады.

Routing
Орналастырылған элементтер арасындағы нақты signal paths FPGA routing ресурстары арқылы құрылады.

Implementation нәтижесіне:
Timing + Congestion + Physical Constraints
әсер етеді.

Мақсат — жобаның физикалық түрде дұрыс орналастырылуын және қажетті design requirements орындалуын қамтамасыз ету.

Timing Constraints және Timing Analysis

Логикалық тұрғыдан дұрыс схема нақты hardware-де қажетті frequency-де де жұмыс істеуі керек.
Сол үшін design tool-ға timing constraints беріледі.

Негізгі мысалдар:
Clock Period / Frequency
Input Delay
Output Delay
Clock Relationships
Timing analysis жобадағы timing paths-ті тексереді.

Негізгі көрсеткіштердің бірі:
Slack
Positive Slack → осы timing check үшін талап орындалған
Negative Slack → осы timing check үшін violation бар

DRC және Design Reports

FPGA design tool әртүрлі есептер қалыптастырады. Оларды тек Success деген мәртебеге қарап бағалау жеткіліксіз.

Негізгі reports:
Synthesis Report — RTL-дің hardware ресурстарына түрленуін бағалау.
Utilization Report — FPGA ресурстарының қолданылуын тексеру.
Timing Report — timing paths және slack мәндерін талдау.
DRC Report — design rules бұзылуын анықтау.
Implementation Report — placement және routing нәтижелерін бағалау.

Warnings, critical messages және **missing constraints** сияқты мәселелерді де зерттеу қажет.

report name	purpose

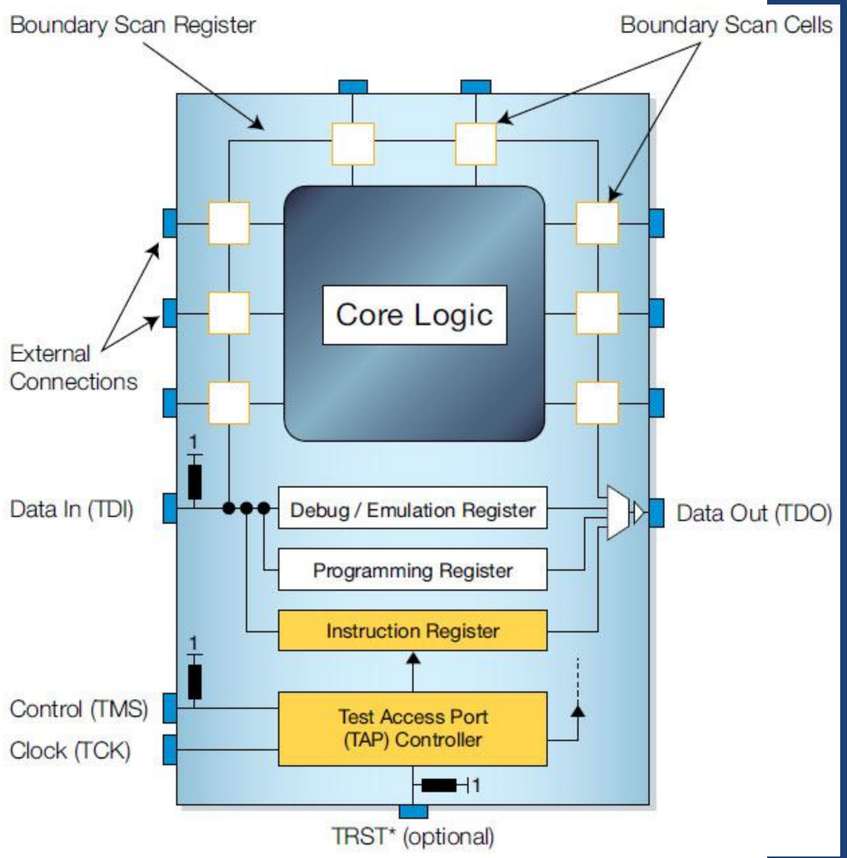
Bitstream және FPGA конфигурациясы

Implementation және **қажетті тексерулер** аяқталғаннан кейін FPGA-ға жүктелетін конфигурациялық деректер қалыптастырылады.

AMD/Xilinx экожүйесінде бұл процесс:
Bitstream Generation
деп аталады.

FPGA конфигурациясын әртүрлі тәсілдермен жүктеуге болады:
JTAG
SPI Flash
немесе құрылғыға байланысты басқа **boot/configuration** әдістері.

Конфигурацияның сәтті жүктелуі жобаның функционалдық тұрғыдан толық дұрыс екенін автоматты түрде білдірмейді.

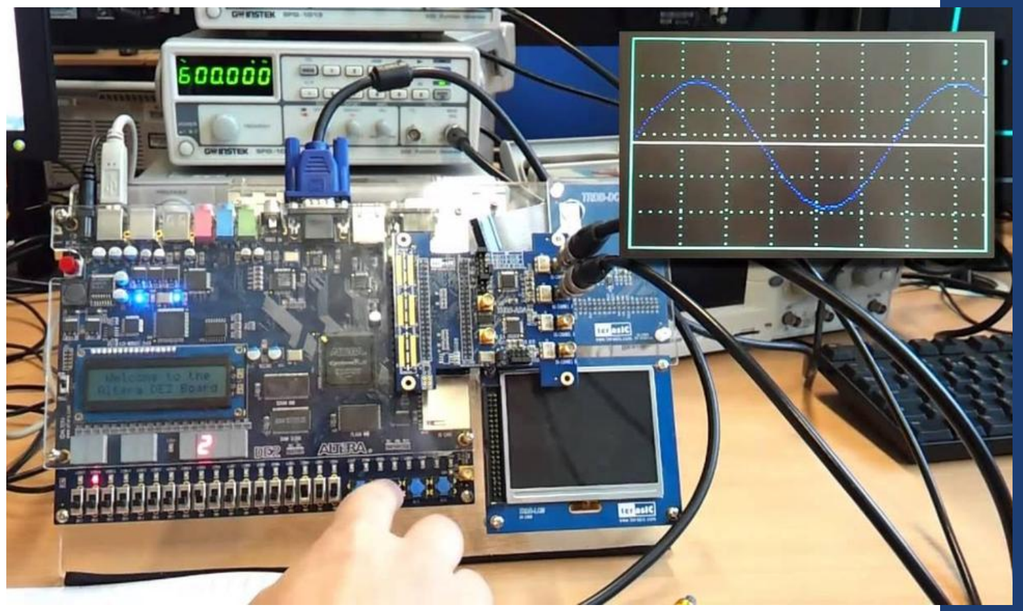


Hardware Verification және Debug

Hardware Verification — дайын FPGA конфигурациясының нақты құрылғыдағы жұмысын тексеру.

Тексерілетін нәрселер:
Clock
Reset
I/O
Interfaces
External Devices
Timing

Debug үшін:
Oscilloscope
Logic Analyzer
Vendor-specific FPGA Debug Tools
қолданылуы мүмкін.

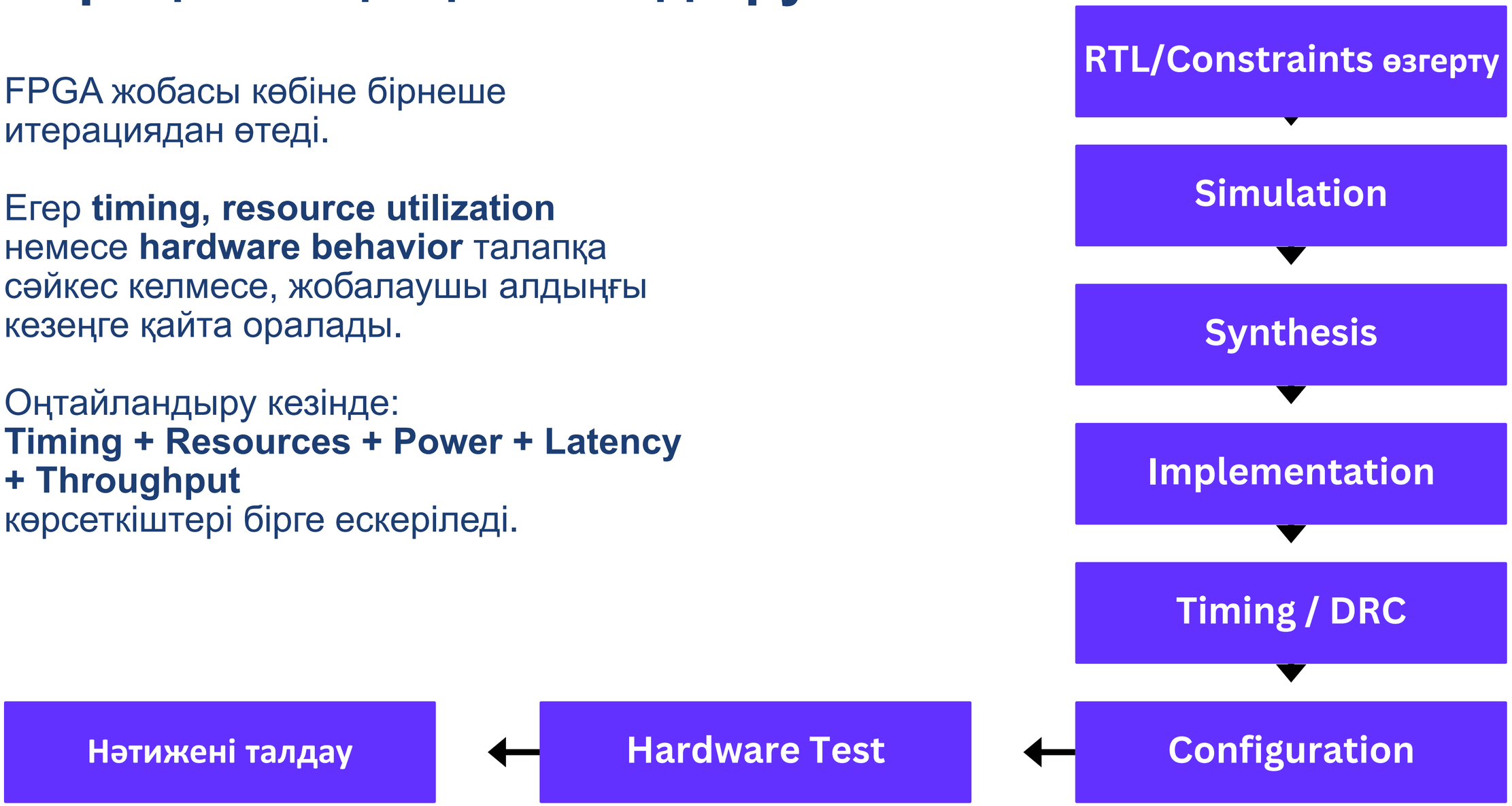


Итерациялық оңтайландыру

FPGA жобасы көбіне бірнеше итерациядан өтеді.

Егер **timing, resource utilization** немесе **hardware behavior** талапқа сәйкес келмесе, жобалаушы алдыңғы кезеңге қайта оралады.

Оңтайландыру кезінде:
Timing + Resources + Power + Latency + Throughput
көрсеткіштері бірге ескеріледі.



Қорытынды

FPGA жобасын әзірлеу бірнеше өзара байланысты кезеңнен тұрады:

- **Specification → RTL → Verification → Synthesis → Implementation → Configuration → Hardware**
- **Simulation** функционалдық қателерді ерте анықтайды.
- **Synthesis** HDL сипаттамасының аппараттық ресурстарға қалай түрленетінін көрсетеді.
- **Implementation** жобаны нақты физикалық FPGA құрылымына орналастырады.
- **Timing Analysis және DRC** жобаның уақыттық және физикалық талаптарын бағалауға мүмкіндік береді.
- **Hardware Verification** дайын жүйенің нақты FPGA құрылғысында жұмысын тексереді.

Сондықтан FPGA design flow — итерациялық инженерлік процесс.

Білімді бекітуге арналған сұрақтар

1.FPGA жобасының толық design flow негізгі қандай кезеңдерден тұрады?
2.Requirements кезеңінде қандай функционалдық және техникалық талаптар анықталады?
3.RTL деңгейінің FPGA жобасындағы рөлі қандай?
4.Top-level module қандай қызмет атқарады?
5.Testbench не үшін қолданылады?
6.Functional simulation мен hardware verification арасындағы айырмашылық қандай?
7.Synthesis нәтижесінде қандай негізгі аппараттық объект алынады?
8.Synthesis report-тан қандай ресурстарды талдауға болады?
9.Placement және routing бір-бірінен несімен ерекшеленеді?
10. Physical constraints implementation нәтижесіне қалай әсер етеді?
11. Timing constraint не үшін қажет?
12. Slack-тің оң және теріс мәндері нені білдіреді?
13. Timing closure дегеніміз не?
14. DRC report қандай мақсатпен қолданылады?
15. Bitstream немесе конфигурациялық файл не үшін қажет?
16. FPGA-ны JTAG арқылы бағдарламалау дегеніміз не?
17. Неліктен simulation-нің сәтті өтуі hardware-дің міндетті түрде дұрыс жұмыс істейтінін білдірмейді?
18. FPGA жобасын оңтайландыру кезінде қандай көрсеткіштер ескеріледі?

Әдебиеттер тізімі:

1.Brown S., Vranesic Z. Fundamentals of Digital Logic with Verilog Design. McGraw-Hill.

2.Chu P. P. FPGA Prototyping by Verilog Examples. Wiley.

3.Harris D. M., Harris S. L. Digital Design and Computer Architecture. Morgan Kaufmann.

4.Wakerly J. F. Digital Design: Principles and Practices. Pearson.

5.AMD. Vivado Design Suite User Guide: Synthesis (UG901), Version 2026.1. AMD Documentation.

6.AMD. Vivado Design Suite User Guide: Using Constraints (UG903), Version 2026.1. AMD Documentation.

7.AMD. Vivado Design Suite User Guide: Design Methodology (UG949), Version 2026.1. AMD Documentation.

8.AMD. Vivado Design Suite User Guide: Design Analysis and Closure Techniques (UG906). AMD Documentation.

9.Intel. Quartus Prime Documentation and Design Recommendations. Intel Documentation.