Жүйелерді FPGA негізінде жобалау

Дәріс 4: Аппараттық құралдарды сипаттау тілдері (HDL)

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

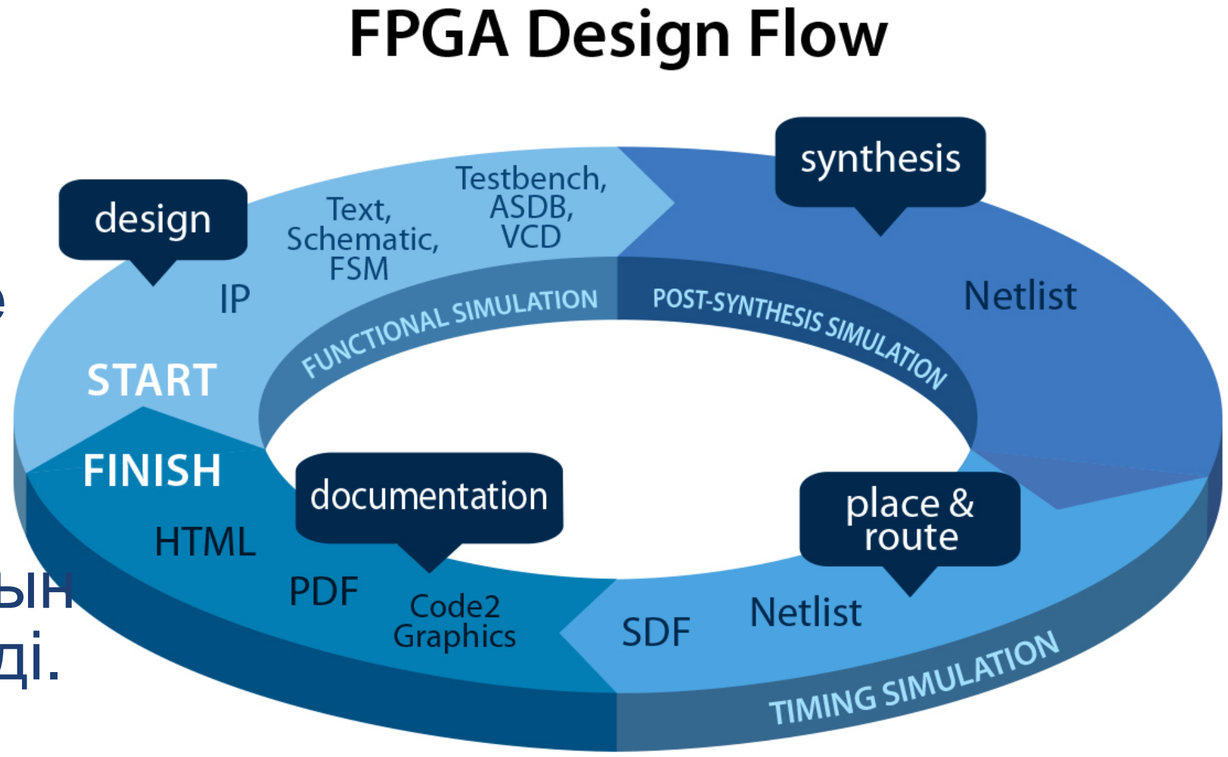
Дәрістің мақсаты:

Аппараттық құралдарды сипаттау тілдерінің (HDL) негізгі қағидаларын меңгеру, HDL тілдерінің цифрлық жүйелерді жобалаудағы рөлін түсіндіру, Verilog, SystemVerilog және VHDL тілдерінің негізгі ерекшеліктерімен таныстыру және аппараттық құрылымды HDL көмегімен сипаттаудың базалық принциптерін қарастыру.

HDL ұғымы және оның цифрлық жобалаудағы рөлі

HDL (Hardware Description Language) — цифрлық аппараттық жүйелердің құрылымын, жұмыс істеу тәртібін және өзара байланысын сипаттауға арналған тіл. Оның көмегімен жобалаушы модульдердің кірістері мен шығыстарын, сигналдар арасындағы логикалық байланыстарды және сақтау элементтерін сипаттайды.

HDL тілдері күрделі цифрлық жүйелерді жеке логикалық элементтерден қолмен құрастырмай-ақ, **формальды түрде жобалауға** мүмкіндік береді. HDL арқылы жазылған сипаттама арнайы жобалау құралдарында өңделіп, FPGA ішінде іске асырылатын аппараттық құрылымға түрлендіріледі.



HDL мен бағдарламалау тілдерінің айырмашылығы

C, C++ және Python сияқты бағдарламалау тілдері алгоритмдер мен командалардың орындалуын сипаттауға бағытталған. Ал HDL тілдерінің **негізгі міндеті** — аппараттық құрылым мен оның жұмысын сипаттау.

HDL-де бірнеше аппараттық блок бір уақытта жұмыс істей алады. Бұл қасиет аппараттық **параллелизм** деп аталады. Мысалы, бір блок деректерді өңдеп жатқан кезде, екінші блок басқа сигналды тәуелсіз өңдей алады.

Сондықтан HDL кодын жазғанда оның қандай функция орындайтынын ғана емес, синтез нәтижесінде қандай аппараттық құрылым пайда болатынын да ескеру қажет.

Аппараттық сипаттаманың негізгі деңгейлері

Аппараттық жүйелерді сипаттаудың **үш негізгі деңгейі** бар:

- **Behavioral** деңгейі жүйенің кіріс сигналдарына қандай жауап беретінін сипаттайды. Мұнда нақты логикалық элементтердің құрылымы толық көрсетілмеуі мүмкін.
- **RTL** (Register Transfer Level) деңгейі регистрлерді, олардың арасындағы деректер тасымалын және комбинациялық логиканы сипаттайды. Бұл деңгей FPGA жобалауында маңызды орын алады.
- **Gate-level** деңгейі жүйені логикалық қақпалар, триггерлер және олардың байланыстары арқылы көрсетеді. Ол аппараттық құрылымды егжей-тегжейлі сипаттайды.

RTL сипаттамасы синтез кезінде логикалық құрылымға түрленіп, FPGA ресурстарына сәйкестендіріледі.

HDL тілдерінің негізгі түрлері

FPGA жобалауында кең таралған **HDL тілдері** — Verilog, SystemVerilog және VHDL.

- **Verilog** цифрлық аппараттық модульдерді, сигналдарды және олардың арасындағы логикалық байланыстарды сипаттауға арналған.
- **SystemVerilog** Verilog мүмкіндіктерін кеңейтеді. Ол аппараттық дизайнмен қатар модельдеу және верификация механизмдерін де қолдайды.
- **VHDL** электрондық жүйелерді формальды сипаттау, модельдеу, тексеру және синтездеу үшін қолданылады.

Үш тілдің мақсаты ортақ болғанымен, олардың синтаксисі мен типтік жүйесі әртүрлі. Тіл таңдау жоба талаптарына, жобалаушылардың тәжірибесіне және қолданылатын құралдарға байланысты.

HDL жобасының құрылымы: Module, Entity және Port

HDL жобасы бір-бірімен байланысқан аппараттық модульдерден құралады. Verilog және SystemVerilog тілдерінде негізгі құрылымдық бірлік *module*, ал VHDL тілінде компоненттің сыртқы интерфейсі *entity* арқылы беріледі. VHDL-де оның ішкі сипаттамасы architecture бөлімінде жазылады.

Port модульдің сыртқы ортамен байланысын қамтамасыз етеді. Ол кіріс, шығыс немесе екі бағытты интерфейс болуы мүмкін. Signal цифрлық мәндерді тасымалдап, аппараттық компоненттер арасындағы байланысты сипаттайды.

Мысалы, қарапайым модуль екі кіріс сигналын қабылдап, олардың логикалық нәтижесін шығысқа бере алады. Осы элементтерді пайдалану арқылы күрделі цифрлық жүйелерді бірнеше өзара байланысқан модульге бөлуге болады.

HDL-дегі сигналдар және негізгі сипаттау тәсілдері

HDL-де сигналдар ақпаратты модульдер арасында тасымалдау үшін қолданылады. Verilog тілінде *wire* аппараттық байланыстарды сипаттауға пайдаланылады, ал SystemVerilog тілінде *logic* типі кеңінен қолданылады. VHDL тілінде сигналды сипаттау үшін *signal* ұғымы пайдаланылады.
Сигналдармен жұмыс істеу кезінде **логикалық**, **арифметикалық**, **салыстыру** және **биттік операторлар** қолданылады.

Мысалы:

```
assign y = a & b;
```

Бұл өрнек y шығысын a және b кірістерінің **логикалық AND** нәтижесіне тең етеді. Демек, аппараттық сипаттамадағы қарапайым оператордың өзі нақты логикалық байланысты білдіреді.

Continuous Assignment және Procedural Description

HDL-де аппараттық логиканы сипаттаудың негізгі тәсілдерінің бірі — **Continuous Assignment**. Verilog тіліндегі assign операторы сигнал мәнін логикалық өрнек арқылы анықтайды. Ол комбинациялық байланыстарды сипаттауға қолайлы.

Feature	Continuous Assignment	Procedural Assignment
Syntax	assign =	always + ᵉ or < =
Evaluation Trigger	wire / net	reg / logic
Used For	Triggered by logcic	Triggered by sensittivity list
Assignment Types	Combinational logic	ᵾ (blocking) an <= (non-blocking
	Only =	= (blocking) <= (non-blocking)

Екінші тәсіл — **Procedural Description**. SystemVerilog тіліндегі always_comb конструкциясы комбинациялық логиканы сипаттайды, ал always_ff тактілік сигналға байланысты жұмыс істейтін тізбектей логиканы сипаттау үшін қолданылады. Бұл тәсілдер арқылы әртүрлі аппараттық құрылымдарды көрсетуге болады. Дегенмен синтаксисті дұрыс жазумен қатар, әр конструкцияның қандай аппараттық ресурсқа айналатынын түсіну маңызды.

Параметрлеу, иерархиялық жобалау және FPGA-ға іске асыру

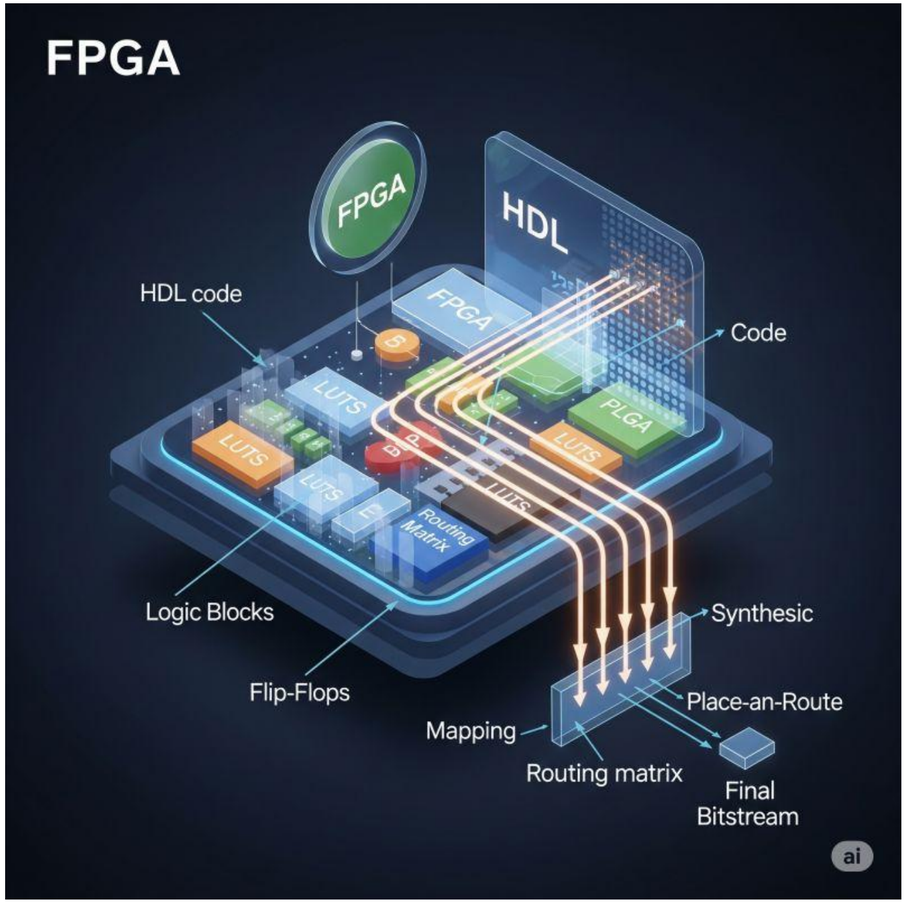
Параметрлеу аппараттық модульдің өлшемдерін немесе жұмыс параметрлерін өзгертіп, оны бірнеше жобада қайта пайдалануға мүмкіндік береді. Verilog және SystemVerilog тілдерінде parameter, ал VHDL тілінде generic механизмі қолданылады.

Иерархиялық жобалау кезінде күрделі жүйе басқару, деректерді өңдеу, жад және интерфейс сияқты функционалдық модульдерге бөлінеді. Бұл жобаның құрылымын түсінуді, жеке блоктарды тексеруді және өзгерістер енгізуді жеңілдетеді.

HDL сипаттамасын FPGA-да іске асырудың **негізгі кезеңдері:**

HDL сипаттамасы → Elaboration → Synthesis → Logic Netlist → Implementation → Конфигурациялау.

Синтез HDL сипаттамасын логикалық құрылымға түрлендіреді. Implementation кезінде құрылым FPGA ресурстарына орналастырылып, байланыстары анықталады. Соңында дайын конфигурация FPGA құрылғысына жүктеледі.



Білімді бекітуге арналған сұрақтар:

1. HDL дегеніміз не және оның негізгі мақсаты қандай?
2. HDL қандай цифрлық жүйелерді сипаттау үшін қолданылады?
3. HDL мен дәстүрлі бағдарламалау тілдерінің негізгі айырмашылығы неде?
4. HDL-дегі аппараттық параллелизм дегеніміз не?
5. Behavioral, RTL және gate-level сипаттамаларының айырмашылығы қандай?
6. Неліктен RTL деңгейі FPGA жобаларында маңызды?
7. Verilog қандай мақсатта қолданылады?
8. SystemVerilog дегеніміз не?
9. VHDL тілінің негізгі ерекшеліктері қандай?
10. Module және Entity ұғымдарының айырмашылығы неде?
11. Port дегеніміз не?
12. Signal және Net қандай қызмет атқарады?
13. Continuous assignment дегеніміз не?
14. Procedural description дегеніміз не?
15. Parameter және Generic не үшін қажет?
16. Иерархиялық жобалау дегеніміз не?
17. HDL сипаттамасы синтез кезінде қалай өңделеді?
18. HDL кодын жазғанда оның аппараттық мағынасын түсіну неліктен маңызды?

Әдебиеттер тізімі:

1. IEEE. IEEE Standard for SystemVerilog — Unified Hardware Design, Specification, and Verification Language. IEEE Std 1800-2023.
2. IEEE. IEEE Standard for VHDL Language Reference Manual. IEEE Std 1076.
3. AMD. Vivado Design Suite User Guide: Synthesis (UG901).
4. AMD. Verilog Functionality – Vivado Design Suite User Guide: Synthesis.
5. AMD. SystemVerilog Support – Vivado Design Suite User Guide: Synthesis.
6. Brown S., Vranesic Z. Fundamentals of Digital Logic with Verilog Design. McGraw-Hill.
7. Harris D., Harris S. Digital Design and Computer Architecture. Morgan Kaufmann.
8. Chu P. P. FPGA Prototyping by Verilog Examples. Wiley.
9. Wakerly J. F. Digital Design: Principles and Practices. Pearson.