Жүйелерді FPGA негізінде жобалау

Дәріс 7: Тактілік сигналдар, синхрондау және FPGA басқару

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

Дәрістің мақсаты:

FPGA негізіндегі синхронды цифрлық жүйелерде тактілік сигналдардың рөлін түсіндіру, clock distribution, reset және synchronization принциптерін қарастыру, metastability мен Clock Domain Crossing (CDC) мәселелерін түсіндіру және timing параметрлерінің сенімді аппараттық жүйе құрудағы маңызын көрсету.

Міндеттері: Clock frequency және period байланысын түсіну; synchronous design принциптерін қарастыру; setup, hold, skew және jitter ұғымдарын меңгеру; reset түрлерін ажырату; metastability және CDC мәселелерін түсіну; Clock Enable және timing constraints рөлін қарастыру.

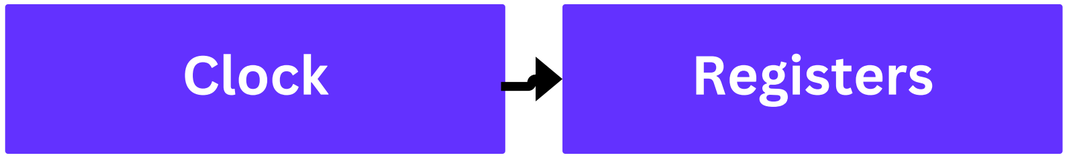
Тактілік сигналдың цифрлық жүйедегі рөлі

Clock (тактілік сигнал) – синхронды цифрлық жүйедегі операциялардың уақыттық ретін ұйымдастыратын периодты басқару сигналы.

Register және **Flip-Flop** сияқты сақтау элементтері clock-тың белгілі бір фронтында өз күйін жаңартады.

Көп жағдайда:
Rising Edge ↑
немесе
Falling Edge ↓
қолданылады.

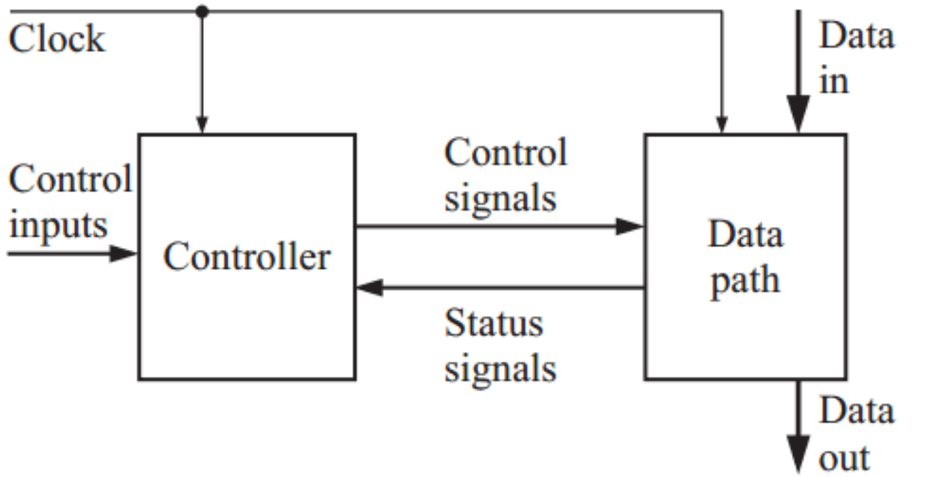
Clock сигналын цифрлық жүйенің уақыттық эталоны ретінде қарастыруға болады.



Синхронды datapath

Синхронды цифрлық жүйенің қарапайым құрылымы:

- Register → Combinational Logic → Register
- Бірінші register деректерді сақтайды.
- Clock оқиғасынан кейін дерек комбинациялық логикаға беріледі.
- Комбинациялық логика есептеуді орындайды.
- Келесі clock оқиғасында екінші register нәтижені қабылдайды.
- Бұл принцип FPGA-дағы синхронды datapath құрылымдарының негізін құрайды



Clock Frequency және Clock Period

Clock frequency (f) – бір секундтағы clock периодтарының саны.
Clock period (T) – екі көршілес бірдей clock фронттарының арасындағы уақыт.

Негізгі байланыс:
T = 1 / f
Мысалдар:
25 MHz → 40 ns
50 MHz → 20 ns
100 MHz → 10 ns
200 MHz → 5 ns

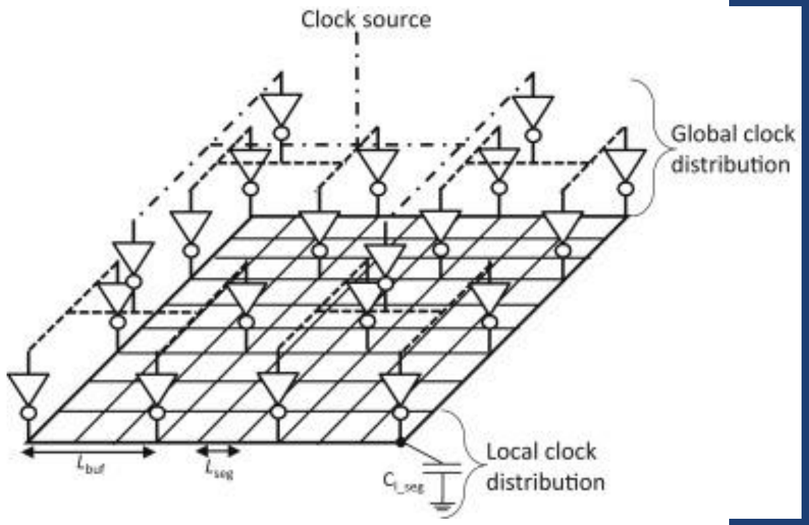
Frequency артқан сайын бір clock cycle ішіндегі қолжетімді уақыт азаяды және timing талаптары қатаңдайды.

FPGA-дағы Clock Distribution

FPGA ішінде бір clock signal көптеген register және Flip-Flop элементтеріне таралуы мүмкін.
Сондықтан clock сигналын қарапайым general-purpose routing арқылы тарату тиімсіз болуы мүмкін.

FPGA архитектурасында clock үшін арнайы:
Clock Source
↓
Clock Management
↓
Dedicated Clock Routing
↓
Clock Regions / Registers
ресурстары қолданылады.

Clock distribution жүйесінің мақсаты — clock signal-ды қабылдаушы элементтерге бақыланатын кідіріспен жеткізу және skew-ті шектеу.



Синхронды цифрлық жобалау

Синхронды дизайнда негізгі state-holding elements анықталған clock сигналдарының бақылауымен жұмыс істейді.

Негізгі принцип:
Clock → State Update

Комбинациялық логика екі clock оқиғасының арасында есептеуді орындайды, ал register келесі clock edge кезінде нәтижені қабылдайды.

Қажетсіз clock сигналдарын логикалық операциялар арқылы құрудан аулақ болған дұрыс.

Мысалы, clock-ты қарапайым AND қақпасымен басқару glitch немесе күтпеген clock edge туғызуы мүмкін.

Setup Time және Hold Time

Register немесе Flip-Flop clock edge келген кезде data сигналын сенімді қабылдауы үшін белгілі уақыттық талаптар орындалуы керек.

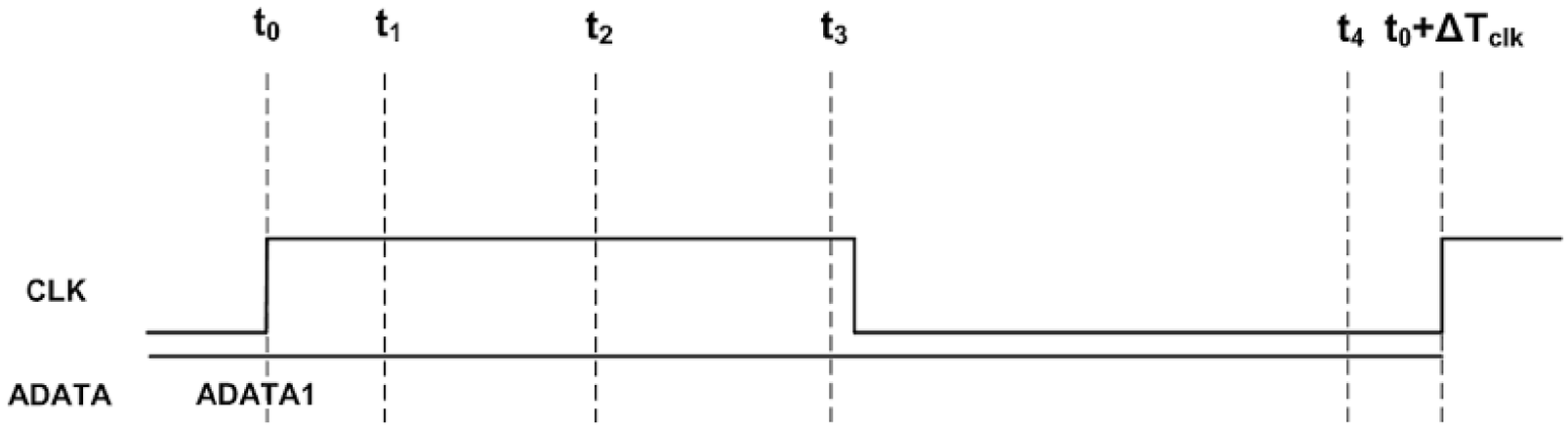
Setup time (tsetup)

Data clock edge келгенге дейін белгілі минималды уақыт бойы тұрақты болуы тиіс.

Hold time (thold)

Data clock edge өткеннен кейін де белгілі минималды уақыт бойы тұрақты болуы тиіс.

Егер setup немесе hold талабы бұзылса, register қате мән қабылдауы немесе metastable күйге түсуі мүмкін.



Clock Skew және Clock Jitter

Clock Skew — бір clock signal-дың әртүрлі register-лерге жету уақыттарының айырмашылығы.

Мысалы:
Register A → 1.0 ns
Register B → 1.2 ns
Skew = 0.2 ns

Clock Jitter — clock edge-тің күтілетін идеалды уақыттан ауытқуы.

Skew негізінен әртүрлі физикалық нүктелердегі clock arrival айырмашылығын сипаттайды, ал jitter clock уақытындағы ауытқуды сипаттайды.

Reset архитектурасы

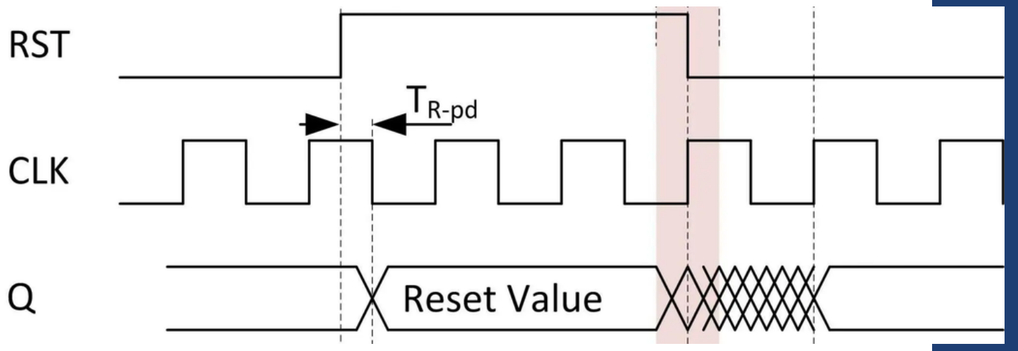
Reset — цифрлық жүйені алдын ала анықталған бастапқы күйге келтіретін басқару сигналы.

Synchronous Reset
Reset тек белгілі clock edge кезінде әсер етеді.

Asynchronous Reset
Reset clock-қа тәуелсіз түрде сақтау элементінің күйіне әсер ете алады.

Практикада **asynchronous assertion + synchronous deassertion** тәсілі жиі қолданылады.

Reset deassertion кезінде clock domain-мен уақыттық қатынасты дұрыс ұйымдастыру маңызды. Сонымен қатар recovery және removal timing талаптары ескеріледі.



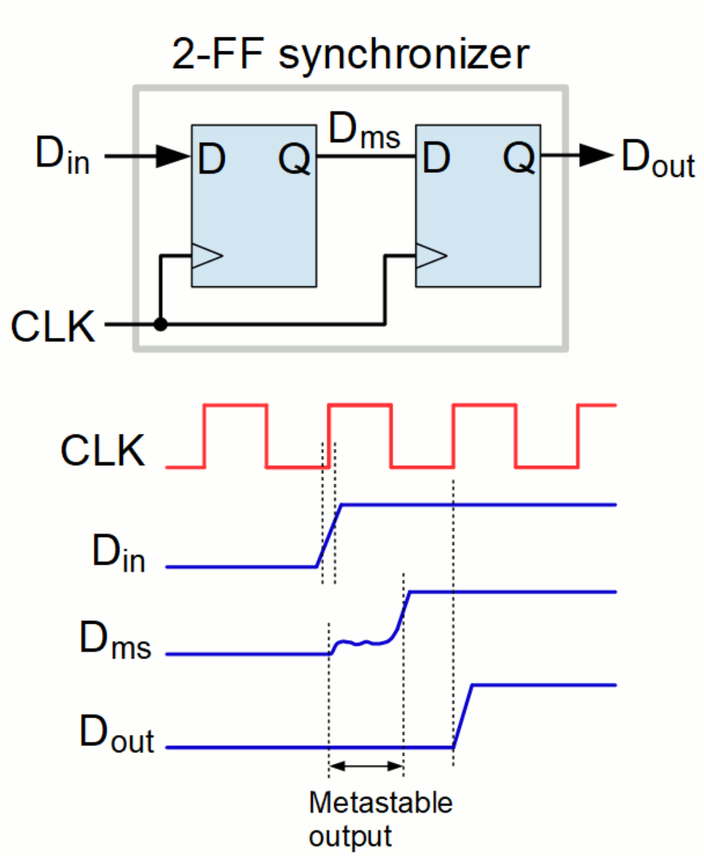
Synchronization және Metastability

Егер асинхронды сигнал clock edge маңында өзгерсе және setup/hold талаптарын бұзса, Flip-Flop **metastable** күйге түсуі мүмкін.

Metastability кезінде output уақытша тұрақсыз болады және бұл жағдай кейінгі логикаға таралуы мүмкін.

Бірбиттік сигналдарды синхрондаудың кең таралған тәсілі:
Async Signal → FF1 → FF2 → Synchronized Signal

Екінші Flip-Flop metastable күйдің келесі логикаға таралу ықтималдығын азайтуға мүмкіндік береді.



Clock Domain Crossing (CDC)

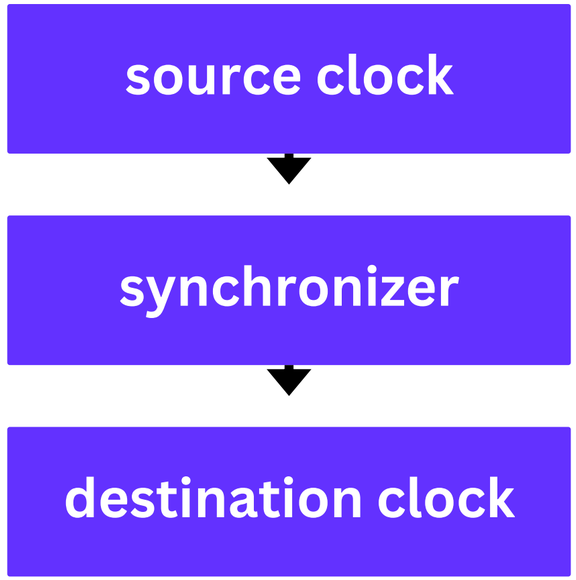
CDC (Clock Domain Crossing) — бір clock domain-де пайда болған сигналды басқа clock domain-де пайдалану немесе деректерді беру процесі.

Егер clock domain-дер арасындағы timing relationship анық болса, олар related/synchronous болуы мүмкін.

Ал фазалық қатынас белгісіз немесе тәуелсіз clock domain-дер арасында сигналды тікелей беру қауіпсіз емес.

1-bit control signal:
Source → Synchronizer → Destination

Multi-bit data:
handshake, asynchronous FIFO немесе арнайы CDC архитектуралары қолданылуы мүмкін.



Clock Enable және Timing Constraints

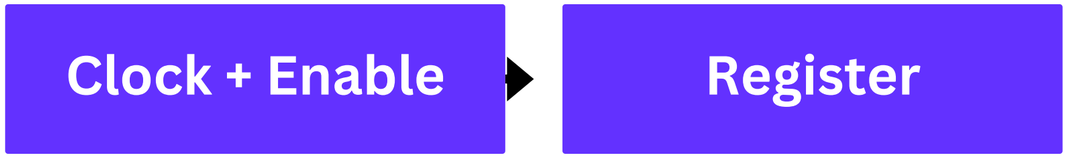
Clock Enable register-дің әр clock edge сайын емес, белгілі шарт орындалғанда ғана жаңартылуына мүмкіндік береді.

Мысалы:
enable = 1 → Q жаңартылады
enable = 0 → Q сақталады

Clock Enable қолданудың маңызды артықшылығы — қарапайым логикалық қақпалар арқылы жаңа clock құру қажеттілігін азайту.

Timing constraints design tool-ға clock period, input/output delay және басқа уақыттық талаптарды береді.

Осы constraints кейін timing analysis кезінде қолданылады.



Қорытынды

FPGA жүйесінің сенімді жұмысы тек логикалық функцияның дұрыстығына емес, оның уақыт бойынша дұрыс ұйымдастырылуына да байланысты.

Негізгі ұғымдар:
Clock → Synchronization → Reset → Timing

- Setup/hold талаптары сақтау элементтерінің сенімді жұмысын анықтайды.
- Skew және jitter clock сигналдарының уақыттық ерекшеліктерін сипаттайды.
- Synchronizer және CDC архитектуралары әртүрлі clock domain-дер арасында қауіпсіз сигнал алмасуға көмектеседі.
- Clock Enable синхронды архитектураны сақтай отырып, register update-ті басқаруға мүмкіндік береді.

Білімді бекітуге арналған сұрақтар

1.Clock сигналы синхронды цифрлық жүйеде қандай қызмет атқарады?
2.Clock frequency мен clock period арасында қандай байланыс бар?
3.100 MHz және 200 MHz clock сигналдарының периодтары қанша?
4.FPGA-да clock үшін неге арнайы routing ресурстары қолданылады?
5.Synchronous design дегеніміз не?
6.Setup time және hold time ұғымдарының айырмашылығы қандай?
7.Clock skew дегеніміз не және ол timing-ке қалай әсер етеді?
8.Clock jitter дегеніміз не?
9.Synchronous reset пен asynchronous reset арасындағы негізгі айырмашылық қандай?
10. Metastability дегеніміз не және ол қандай жағдайда пайда болуы мүмкін?
11. Екі flip-flop synchronizer қандай мәселені шешуге көмектеседі?
12. Неліктен multi-bit CDC үшін әр битке жеке synchronizer жеткіліксіз болуы мүмкін?
13. Clock Enable қандай мақсатта қолданылады?
14. Неліктен clock сигналын қарапайым логикалық қақпалар арқылы gate ету ұсынылмайды?
15. Timing constraint не үшін қажет?
16. Оң және теріс slack нені білдіреді?
17. CDC мен timing analysis арасында қандай байланыс бар?
18. Clock және reset логикасын simulation кезінде қандай жағдайлармен тексеру керек?

Әдебиеттер тізімі:

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

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

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

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

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

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

7.AMD. Vivado Design Suite User Guide: Synthesis (UG901). AMD Documentation.

8.Intel. Quartus Prime Pro Edition User Guide: Design Recommendations. Intel Documentation.