Жүйелерді FPGA негізінде жобалау

Дәріс 5: Комбинациялық және тізбектей жұмыс істейтін схемаларды жобалау

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

Дәрістің мақсаты:

Комбинациялық және тізбектей жұмыс істейтін цифрлық схемаларды жобалау принциптерін түсіндіру, олардың құрылымдық айырмашылықтарын көрсету және FPGA негізіндегі жүйелерде мұндай схемаларды HDL арқылы іске асырудың негізгі тәсілдерін қарастыру.

Міндеттері: Комбинациялық және тізбектей схемалардың ерекшеліктерін түсіну, MUX, decoder, encoder, comparator, adder сияқты комбинациялық блоктарды және flip-flop, register, shift register, counter сияқты тізбектей схемаларды қарастыру, сондай-ақ олардың FPGA жобаларында қолданылуын түсіну.

Схемаларды жобалаудың жалпы принциптері

Цифрлық схеманы жобалау белгілі бір функцияны формальды түрде анықтаудан басталады.

Негізгі кезеңдер:

Талаптар → Кірістер/шығыстар → Функционалдық сипаттама → Аппараттық құрылым → HDL → Simulation/Synthesis

Функционалдық сипаттаманы ақиқаттық кесте, логикалық өрнек немесе блоктық схема түрінде беруге болады.

Негізгі мақсат — HDL кодын ғана емес, оның қандай аппараттық құрылымды сипаттайтынын түсіну.



Комбинациялық схемаларды жобалау

Комбинациялық схемада шығыс мәні тек **ағымдағы кірістерге** тәуелді болады. Мұндай схема өзінің алдыңғы жағдайын сақтамайды.

Мысалдар:
MUX
Decoder
Encoder
Comparator
Adder

Комбинациялық логикада барлық мүмкін басқару жағдайларында шығыстардың анықталған мәні болуы маңызды.

HDL сипаттамасында кейбір жағдайларда шығысқа мән берілмесе, synthesis tool байқаусыз latch инференциялауы мүмкін.

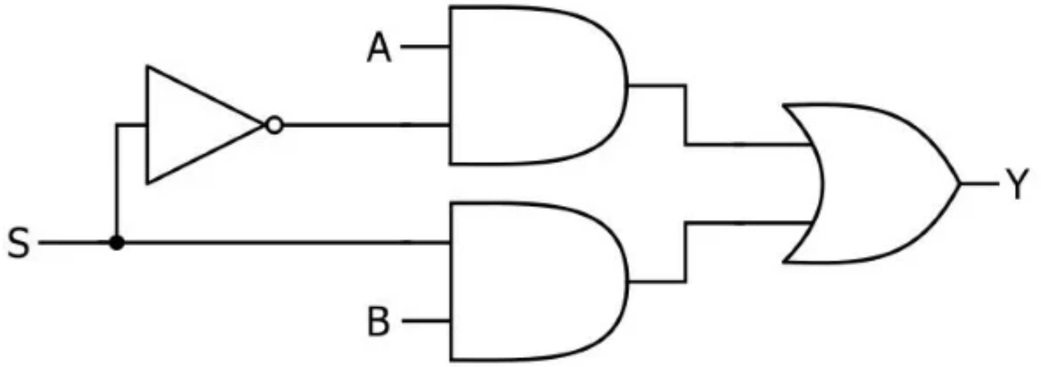
Мультиплексорды жобалау

MUX (Multiplexer) бірнеше кірістің ішінен бір сигналды таңдап, оны шығысқа өткізеді.

2-to-1 MUX:
D0, D1 → MUX → Y

Бір таңдау сигналы S қолданылады:
S = 0 → Y = D0
S = 1 → Y = D1

Логикалық функция:
Y = ¬S·D0 + S·D1
m таңдау сигналы арқылы 2^m кірістің біреуін таңдауға болады.



2-to-1 MUX

MUX-ты HDL арқылы сипаттау

MUX-тың аппараттық функциясын HDL арқылы қысқаша сипаттауға болады:

```
always_comb begin
    if (sel)
        y = d1;
    else
        y = d0;
end
```

Бұл кодтың негізгі мағынасы:
sel = 0 → d0
sel = 1 → d1

Мұнда маңыздысы синтаксисті жаттау емес, HDL сипаттамасының комбинациялық hardware қалыптастыратынын түсіну.

Декодерлер мен кодерлер

Decoder кіріс кодын сәйкес шығыс комбинациясына түрлендіреді.

Мысалы:
2-to-4 Decoder
2 кіріс → 4 ықтимал шығыс

Encoder кері функцияны орындайды:
Белсенді кіріс → Екілік код

Қарапайым encoder-де бір уақытта бір кіріс белсенді деп есептеледі.

Егер бірнеше кіріс қатар белсенді болуы мүмкін болса, Priority Encoder қолданылады.

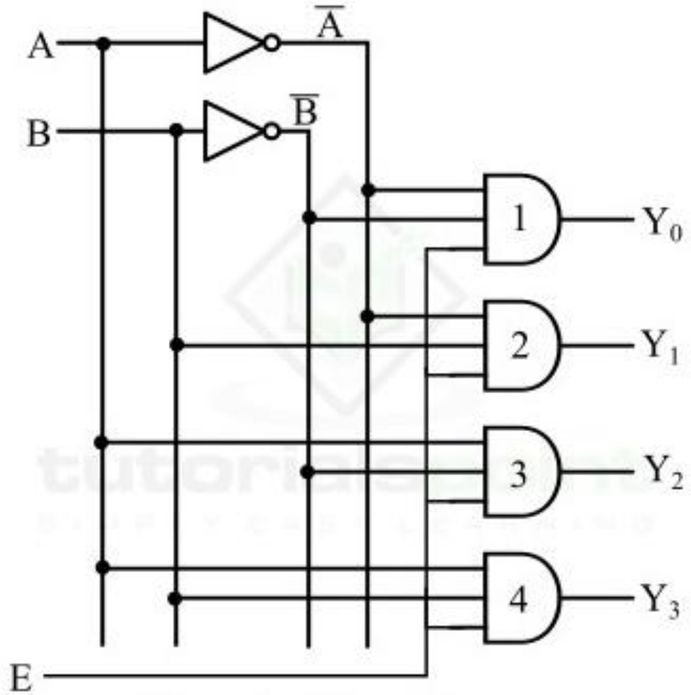
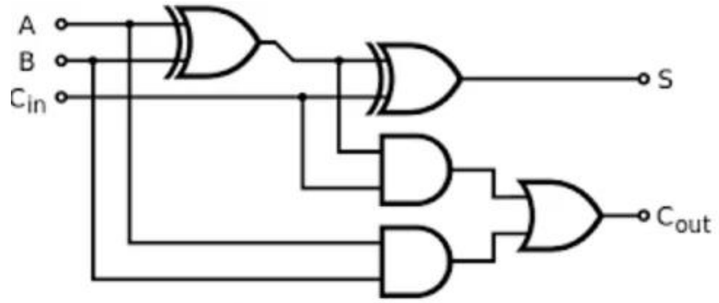


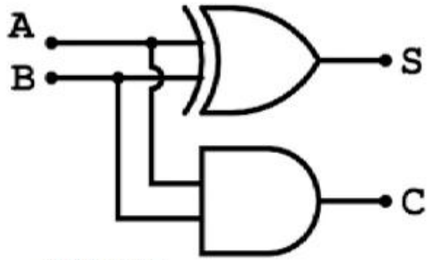
Figure 3 - 2 to 4 Decoder

Компараторлар және сумматорлар

Comparator екі екілік санды салыстырады.
Негізгі нәтижелер:
A > B
A = B
A < B
Half Adder екі битті қосады:
$S = A \oplus B$
$C = A \cdot B$
Full Adder қосымша Carry-in сигналын ескереді:
$S = A \oplus B \oplus Cin$
$Cout = A \cdot B + Cin \cdot (A \oplus B)$



FULL ADDER



HALF ADDER

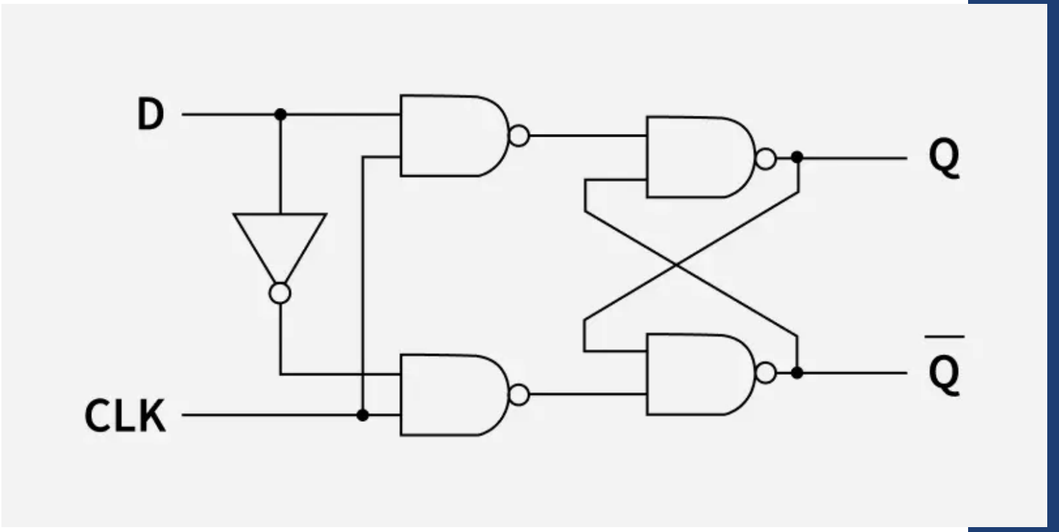
Тізбектей жұмыс істейтін схемаларды жобалау

Тізбектей жұмыс істейтін схема ағымдағы кірістермен қатар алдыңғы күйді де ескереді.

Мұнда сақтау элементтері қолданылады:
Flip-Flop → Register → Counter
Синхронды тізбектей схемада күйдің өзгеруі clock сигналымен байланысты.

D-триггердің қарапайым принципі:
Qnext = D

Бұл принцип регистрлердің, санауыштардың, жылжыту регистрлерінің және кейінгі FSM құрылымдарының негізін құрайды.



Триггерлер, регистрлер және жылжыту регистрлері

Flip-Flop бір бит ақпаратты сақтайды.
Бірнеше flip-flop бірігуі арқылы register құрылады.

Мысалы:
8-bit Register = 8 сақтау элементі
Register деректерді уақытша сақтау және синхронды datapath ұйымдастыру үшін қолданылады.
Shift Register әр clock оқиғасында деректерді көршілес разрядтарға жылжытады.

Мысалы:
Q3 ← SerialIn
Q2 ← Q3
Q1 ← Q2
Q0 ← Q1

Shift register serial/parallel түрлендіру, деректерді кідірту және буферлеу үшін пайдаланылуы мүмкін.

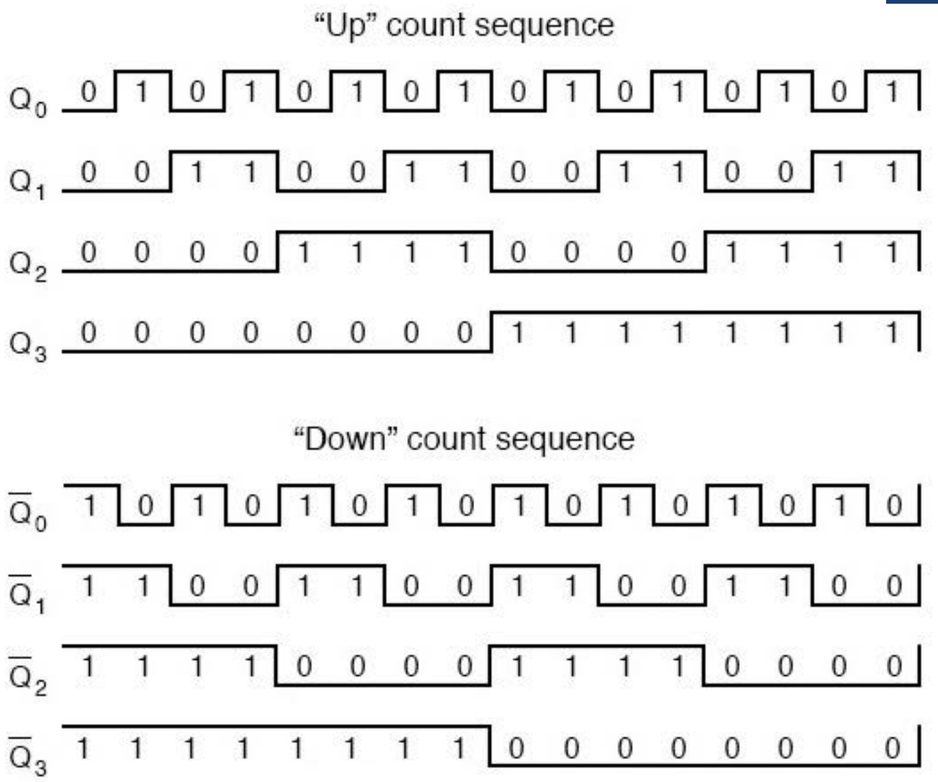
Санауыштарды жобалау

Counter тактілік оқиғаларды санап, өзінің күйін алдын ала анықталған реттілік бойынша өзгертеді.

3-bit up-counter:
000 → 001 → 010 → 011 → 100 → 101 → 110 → 111 → 000

Counter түрлері:
Up-counter — мәнді арттырады.
Down-counter — мәнді азайтады.
Mod-N counter — N күйден кейін бастапқы күйге қайтады.

FPGA жобаларында синхронды counter жиі қолданылады. Enable және Reset сигналдары арқылы оның жұмысын басқаруға болады.



Қорытынды

Комбинациялық схемалар ағымдағы кірістерге тәуелді болса, тізбектей жұмыс істейтін схемалар алдыңғы күйді сақтау мүмкіндігін пайдаланады.

Негізгі комбинациялық блоктар:
MUX → Decoder → Encoder → Comparator → Adder

Негізгі тізбектей блоктар:
Flip-Flop → Register → Shift Register → Counter

Оларды біріктіру арқылы күрделі datapath және басқару жүйелері құрылады.

Білімді бекітуге арналған сұрақтар

1.Комбинациялық схема дегеніміз не және оның шығысы неге тәуелді?
2.Тізбектей жұмыс істейтін схема комбинациялық схемадан несімен ерекшеленеді?
3.Комбинациялық логикада барлық шығыс сигналдарын барлық тармақтарда анықтаудың маңызы неде?
4.2-to-1 мультиплексор қалай жұмыс істейді?
5.Мультиплексордағы таңдау сигналдарының саны мен кірістер саны арасында қандай байланыс бар?
6.Декодер мен кодердің негізгі айырмашылығы қандай?
7.Priority encoder қандай жағдайда қажет болады?
8.Half Adder мен Full Adder арасындағы айырмашылық неде?
9.Компаратор қандай логикалық нәтижелерді қалыптастырады?
10. Триггер қандай қызмет атқарады?
11. Регистр мен санауыштың құрылымдық айырмашылығы неде?
12. Синхронды санауыштың артықшылығы қандай?
13. Clock Enable сигналы не үшін пайдаланылады?
14. Комбинациялық логика мен регистрлерді біріктіру арқылы қандай құрылым алуға болады?
15. Simulation мен synthesis нәтижелерінің негізгі айырмашылығы неде?
16. HDL сипаттамасында latch-тің байқаусыз пайда болуына не себеп болуы мүмкін?

Әдебиеттер тізімі:

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

2. Mano M. M., Ciletti M. D. Digital Design. Pearson.

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

4. Roth C. H., Kinney L. L. Fundamentals of Logic Design. Cengage Learning.

5. Harris D., Harris S. Digital Design and Computer Architecture. Morgan Kaufmann.

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

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

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