ИжМС

Дәуренов ӘД

Дәріс 8. Қосқыштар және арифметикалық құрылғылар

Дәрістің мақсаты

Білім алушыларға цифрлық құрылғылардағы

- арифметикалық операциялардың орындалу принциптерін,
- жартылай қосқыш пен толық қосқыштың құрылысын,
- көпразрядты қосқыштардың жұмысын және
- арифметикалық-логикалық құрылғының (ALU) қызметін түсіндіру.

Дәріс жоспары

1. Цифрлық құрылғылардағы арифметикалық операциялар.
2. Екілік сандарды қосу принципі.
3. Жартылай қосқыш (Half Adder).
4. Толық қосқыш (Full Adder).
5. Көпразрядты қосқыштар.
6. Екілік сандарды азайту.
7. Арифметикалық-логикалық құрылғы (ALU).
8. Қосқыштардың микропроцессорлардағы қолданылуы.

«Интегралдық және микропроцессорлық схемотехника» пәні бойынша алғашқы 7 дәріске шолу

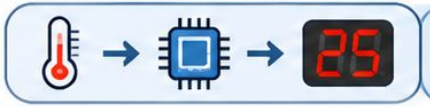
Алғашқы жеті дәрісте біз цифрлық электрониканың негізгі ұғымдарын —
- ақпаратты 0 мен 1 арқылы көрсетуден бастап,
- деректерді өңдейтін және
- түрлендіретін логикалық схемаларды құруға дейін қарастырдық.

Өткен тақырыптардың негізгі мақсаты — цифрлық құрылғының ақпаратты қалай қабылдайтынын, өңдейтінін және қажетті нәтижені қалай қалыптастыратынын түсіну.

1. Біз не үйрендік және бұл не үшін қажет?

№	Тақырып	Не үйрендік?	Не үшін қажет?
1	Санау жүйелері	Екілік, ондық, он алтылық санау жүйелері және сандарды түрлендіру	Компьютерлер мен микроконтроллерлерде ақпараттың қалай көрсетілетінін түсіну үшін
2	Буль алгебрасы	AND, OR, NOT, XOR операциялары және логикалық өрнектер	Цифрлық құрылғылардың жұмыс шарттарын математикалық түрде сипаттау үшін
3	Карно карталары	Логикалық функцияларды минимизациялау	Логикалық элементтер саны аз, қарапайым схемаларды құру үшін
4	TTL және CMOS	Логикалық микросхемалар, TTL және CMOS технологиялары	Логикалық операциялардың электрондық схемаларда физикалық түрде қалай жүзеге асатынын түсіну үшін
5	Шифраторлар және дешифраторлар	Сигналдарды екілік кодқа айналдыру және кодтарды дешифрлау	Ақпаратты кодтау, құрылғыларды таңдау және шығыстарды басқару үшін
6	Мультиплексорлар және демультиплексорлар	Бірнеше сигналдың біреуін таңдау және сигналды қажетті арнаға бағыттау	Деректер ағынын басқару және байланыс арналарын таңдау үшін
7	Код түрлендіргіштері	Бір цифрлық кодты басқа кодқа түрлендіру, мысалы BCD → 7-сегменттік индикатор	Әртүрлі цифрлық құрылғылардың деректерді дұрыс түсінуі және көрсетуі үшін

Цифрлық құрылғыны құрайтын негізгі тақырыптардың байланысы



Бірнеше датчиктен ақпарат қабылдайтын, оны өңдейтін және нәтижені дисплейде көрсететін цифрлық құрылғыны елестетейік.

1 Санау жүйелері

- Ақпарат екілік (0 және 1) кодта беріледі.
- Сандарды және деректерді цифрлық түрде көрсету тәсілдерін үйренеміз.

$13_{10} \rightarrow 1101_2$

2–4 Логика, Карно карталары, TTL/CMOS

- Жұмыс шарттарын анықтаймыз.
- Логикалық функцияларды ықшамдаймыз (Карно карталары).
- Логикалық элементтердің және TTL/CMOS технологияларының ерекшеліктерін түсінеміз.



A\B	0	1
0	0	0
1	1	1



5 Шифраторлар және дешифраторлар

- Ақпаратты кодтаймыз (шифратор) немесе қажетті шығысты таңдаймыз (дешифратор).



6 Мультиплексорлар және демультиплексорлар

- Бірнеше сигналдың ішінен біреуін таңдаймыз (мультиплексор).
- Бір сигналды бірнеше бағытқа таратамыз (демультиплексор).



7 Код түрлендіргіштері

- Деректерді әр түрлі код форматтарына түрлендіреміз.
- Мысалы: екілік код → BCD, BCD → 7-сегмент код.



Нәтиже

Цифрлық құрылғы қажетті функцияны орындайды.

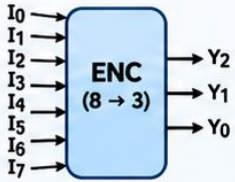
Бұл — тақырыптардың өзара байланысын көрсететін жалпыланған схема. Әрбір цифрлық құрылғыда осы блоктардың барлығы міндетті түрде қолданылады.

Негізгі цифрлық құрылғылар және олардың қызметі

Шифраторларды, дешифраторларды, мультиплексорларды, демультиплексорларды және код түрлендіргіштерін бір-бірінен ажырата білу өте маңызды.

1 Шифратор (Encoder)

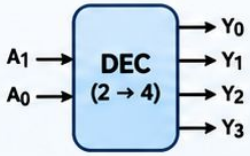
- Белсенді кірістің нөмірін екілік кодқа түрлендіреді.
- **Мысалы:** №5 кіріс белсенді → **101** коды.



Белсенді кіріс	Код (Y2Y1Y0)
I_0	000
I_1	001
I_2	010
I_3	011
I_4	100
I_5	101
I_6	110
I_7	111

2 Дешифратор (Decoder)

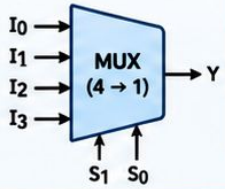
- Екілік кодқа сәйкес келетін шығысты белсендіреді.
- Мысалы: **10** коды → №2 шығыс белсенді болады.



A_1	A_0	Белсенді шығыс
0	0	Y_0
0	1	Y_1
1	0	Y_2
1	1	Y_3

3 Мультиплексор (MUX)

- Бірнеше кіріс сигналының біреуін таңдап, оны ортақ шығысқа жібереді.
- **Мысалы:** төрт датчиктің біреуінен алынған деректер таңдалады.

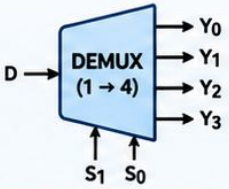


S_1	S_0	Шығыс Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Цифрлық құрылғылар

4 Демультиплексор (DEMUX)

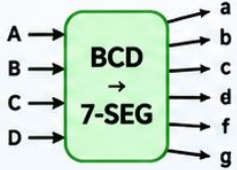
- Бір кіріс сигналын таңдалған шығысқа бағыттайды.
- Мысалы: басқару сигналы төрт арнаның біреуіне жіберіледі.



S_1	S_0	Белсенді шығыс
0	0	Y_0
0	1	Y_1
1	0	Y_2
1	1	Y_3

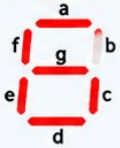
5 Код түрлендіргіші (Code Converter)

- Ақпаратты кодтаудың бір түрін басқа түріне түрлендіреді.
- **Мысалы:** 0101 BCD коды → 5 цифрын көрсету үшін қажетті сегмент сигналдары.



BCD (A B C D)	Цифр	7-сегмент (a b c d e f g)
0101	5	1 0 1 1 0 1 1





Қорытынды

- ✔ **Шифратор** — кіріс нөмірін кодқа түрлендіреді.
- ✔ **Дешифратор** — кодқа сәйкес шығысты белсендіреді.
- ✔ **Мультиплексор** — бірнеше кірістен біреуін таңдайды.
- ✔ **Демультиплексор** — бір кірісті бірнеше арнаға бөледі.
- ✔ **Код түрлендіргіші** — кодтардың түрін өзгертеді (мысалы, BCD → 7-сегмент).

Біз қандай білім мен дағдыларды меңгердік?

Алғашқы жеті тақырыпты оқып болғаннан кейін білім алушы:

- Екілік сандарды оқып, жаза алуы;
- Ақиқат кестелерін құра алуы;
- AND, OR, NOT, XOR операциялары арқылы логикалық функцияларды жаза алуы;
- Карно карталарының көмегімен логикалық функцияларды ықшамдай алуы;
- Логикалық элементтерді ажыратып, TTL және CMOS технологияларының ерекшеліктерін түсінуі;
- Шифраторлардың, дешифраторлардың, мультиплексорлардың және демультиплексорлардың жұмысын түсіндіре алуы;
- Қарапайым код түрлендіргіштерін жобалай алуы;
- CircuitVerse ортасында комбинациялық логикалық схемаларды құрып, тексере алуы тиіс.

Бұл білімдер микропроцессорларды әрі қарай зерттеу үшін неге маңызды?

Осы уақытқа дейін біз негізінен комбинациялық цифрлық схемаларды зерттедік. Мұндай схемалардың шығыс мәндері кірістердің ағымдағы мәндеріне байланысты анықталады.
Енді күрделірек құрылғыларды зерттеуге көше аламыз.
Мысалы, келесі «Қосқыштар және арифметикалық құрылғылар» тақырыбында XOR, AND және OR логикалық элементтерінен екілік сандарды қосатын құрылғыны қалай құруға болатынын үйренеміз.
Одан кейін триггерлерді, санауыштарды және регистрлерді қарастырамыз. Бұл құрылғылар алдыңғы күйлер туралы ақпаратты сақтай алады, яғни жады қызметін атқарады.
Болашақта осы білімдер микропроцессорлар мен микроконтроллерлердің архитектурасын түсінуге көмектеседі.

1. Цифрлық құрылғылардағы арифметикалық операциялар

Компьютерлер, микроконтроллерлер және басқа да цифрлық құрылғылар **ақпаратты екілік код** түрінде өңдейді.
Екілік жүйеде тек екі сан қолданылады: 0 және 1
Сондықтан цифрлық құрылғылардағы арифметикалық операциялар логикалық элементтердің көмегімен орындалады.
Негізгі арифметикалық операциялар:

- Қосу — Addition
- Азайту — Subtraction
- Көбейту — Multiplication
- Бөлу — Division

Осы операциялардың ішінде **қосу – ең негізгі операциялардың бірі** болып саналады.
Себебі көптеген арифметикалық есептеулер қосу операциясына негізделеді. Мысалы, азайтуды қосымша код арқылы қосу түрінде орындауға болады.

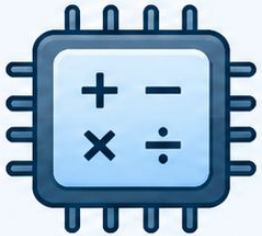
Цифрлық құрылғыда арифметикалық есептеудің орындалуы

Екілік деректер (A, B)

A = 0101_2 (5_{10})

B = 0011_2 (3_{10})

Арифметикалық құрылғы (ALU)



- Қосу
- Азайту
- Көбейту
- Бөлу
- Басқа операциялар (мысалы, салыстыру)

Нәтиже (екілік код)

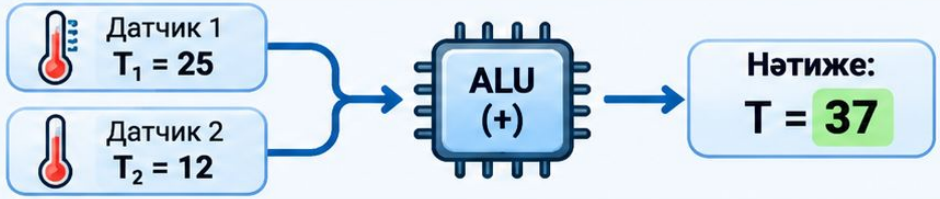
$A + B = 1000_2$ (8_{10})

Арифметикалық құрылғы не істейді?



Екілік кодпен берілген сандарға арифметикалық операцияларды (қосу, азайту, көбейту, бөлу және т.б.) орындайды.

Мысал: микроконтроллерде екі датчик мәнін қосу



Яғни, микроконтроллер екі датчиктің көресткіштерін қосуы немесе өлшенген мән мен берілген мәннің айырмасын есептеуі мүмкін.

2. Екілік сандарды қосу принципі

Екілік сандарды қосу ондық сандарды қосуға ұқсас. Бірақ екілік жүйеде тек 0 және 1 цифрлары қолданылады.

Екілік қосу ережелері

A	B	Қосынды (S)	Тасымал (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Негізгі ережелер

- 0 + 0 = 0 (тасымал жоқ)
- 0 + 1 = 1 (тасымал жоқ)
- 1 + 0 = 1 (тасымал жоқ)
- **1 + 1 = 0, тасымал 1**

1 + 1 → S = 0, C = 1

Мысал: 5 + 3

$5_{10} = 0101_2$

$3_{10} = 0011_2$

Екілік қосу

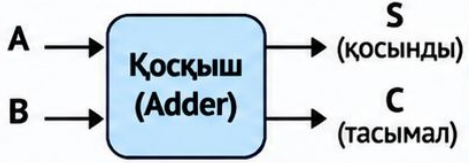
$$\begin{array}{r} 0\ 1\ 0\ 1 \\ +\ 0\ 0\ 1\ 1 \\ \hline 1\ 0\ 0\ 0 \end{array}$$

$0101_2 + 0011_2 = 1000_2 = 8_{10}$

Қосу процесі (оңнан солға)

1. 1 + 1 = 0, тасымал 1
2. 0 + 1 + 1 = 0, тасымал 1
3. 1 + 0 + 1 = 0, тасымал 1
4. 0 + 0 + 1 = 1, тасымал жоқ

Екілік сандарды қосу схемасы



Қорытынды

- Қосындының әр разряды 0 немесе 1 болады.
- Келесі (жоғары) разрядқа тасымал 1 болғанда беріледі.

Бұл операцияны цифрлық схемада орындау үшін арнайы құрылғы — **қосқыш (Adder)** пайдаланылады.

3. Жартылай қосқыш (Half Adder)

Жартылай қосқыш — екі бірлік разрядты (екілік сандардың) **қосуға арналған** комбинациялық логикалық құрылғы.

Кіріс және шығыс сигналдары

- **Кіріс:** A, B — қосылатын екілік разрядтар.
- **Шығыс:** S — қосынды, C — тасымал.

Жартылай қосқыштың шартты белгісі



Ақиқат кестесі

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

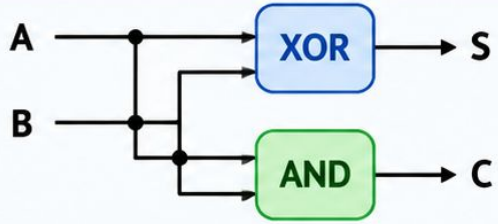
Қосынды және тасымал формулалары

$$S = A \oplus B$$

$$C = A \cdot B$$

- Қосынды S кірістердің мәндерінің тақ (әртүрлі) болған жағдайында 1 болады.
- Тасымал C екі кіріс те 1 болған кезде 1 болады.

Жартылай қосқыштың логикалық схемасы



- S (қосынды) — XOR (немесе айрықша «НЕМЕСЕ») элементі арқылы алынады.
- C (тасымал) — AND (логикалық «ЖӘНЕ») элементі арқылы алынады.

Жұмыс істеу мысалы

Мысалы: 1 + 1

A = 1, B = 1

- $S = A \oplus B = 1 \oplus 1 =$ **0**
- $C = A \cdot B = 1 \cdot 1 =$ **1**

Қорытынды

- ✔ **Жартылай қосқыш** екі бірілік разрядты **қосады.**
- ✔ Нәтижесінде екі сигнал береді: **S** — қосынды және **C** — тасымал.

Жартылай қосқыш (Half Adder) неге жеткіліксіз?

Жартылай қосқыш тек екі битті қоса алады: A+B
Мысалы: 1+1=10
Мұндағы:

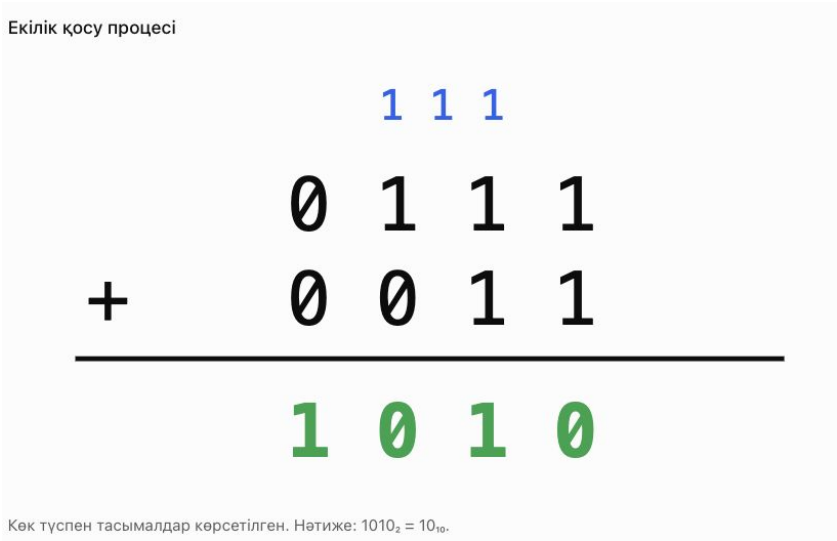
- 0 — қосынды биті (S).
- 1 — келесі разрядқа берілетін тасымал (Carry).

Яғни жартылай қосқыш дұрыс жұмыс істейді.
Бірақ оның тек екі кірісі бар: A және B.

Үшінші бит қайдан пайда болады?

Мысалы, екі екілік санды қосайық $0111_2 + 0011_2$

Бұл ондық жүйеде 7+3=10.



Енді оң жақтан бастап есептейік.

Бірінші разряд:

$$1 + 1 = 10_2$$

Қосындыға 0 жазамыз, ал 1 келесі разрядқа өтеді.

Екінші разряд:

Екі санның келесі биттері де 1-ге тең. Бірақ алдыңғы разрядтан тағы бір 1 келді!

Сондықтан:

$$\underbrace{1+1}_{\text{екі кіріс}} + \underbrace{1}_{\text{алдыңғы разрядтан тасымал}} = 11_2$$

Міне, үш битті қосу қажеттілігі осы жерде пайда болады.

Сондықтан толық қосқыш (Full Adder) керек

Толық қосқыш (Full Adder)

Толық қосқыш — үш бірразрядты екілік мәнді қосатын комбинациялық логикалық құрылғы.

Оның үш кірісі бар:

- A — бірінші бит.
- B — екінші бит.
- Cin — алдыңғы разрядтан келетін тасымал.

Екі шығысы бар:

- S — қосынды.
- Cout — келесі разрядқа берілетін тасымал.

Толық қосқыш (Full Adder)

Толық қосқыш — үш бірразрядты екілік мәнді қосатын комбинациялық логикалық құрылғы.

Кіріс сигналдары

- A — бірінші бит.
- B — екінші бит.
- C_{in} — алдыңғы разрядтан келетін тасымал.

Шығыс сигналдары

- S — қосынды.
- C_{out} — келесі разрядқа берілетін тасымал.

Толық қосқыштың шартты белгісі



Толық қосқыштың ақиқат кестесі

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	**1**	**1**	**1**	**1**

Логикалық формулалар

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = A \cdot B + (A \oplus B) \cdot C_{in}$$

Қысқаша түсіндіру

- S — үш биттің тақ санды (1, 3) болған жағдайында 1 болады.
- C_{out} — кемінде екі кіріс 1 болған жағдайында 1 болады.

Қорытынды

- ✓ Толық қосқыш үш бірразрядты екілік сандарды қосады.
- ✓ Нәтижесінде екі сигнал береді: S — қосынды және C_{out} — тасымал.

Толық қосқышты екі жартылай қосқышпен және OR элементімен құрастыру

Толық қосқышты екі Half Adder және бір OR элементі арқылы жүзеге асыруға болады.

Жартылай қосқыштардың теңдеулері

1 Бірінші жартылай қосқыш:

$S_1 = A \oplus B$ $\quad C_1 = AB$

2 Екінші жартылай қосқыш:

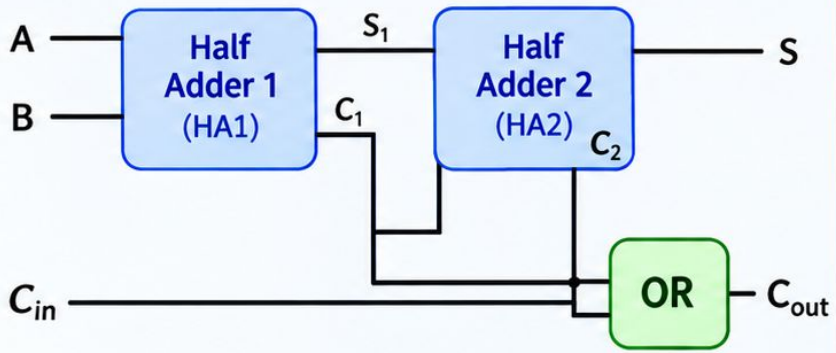
$S = S_1 \oplus C_{in}$ $\quad C_2 = S_1 C_{in}$

Шығыс тасымалды есептеу

Екі тасымалды біріктіреміз:

$$C_{out} = C_1 + C_2$$

Толық қосқыштың құрылымдық схемасы



Элементтер

- **HA1** — A және B кірістерін қосады, S_1 және C_1 береді.
- **HA2** — S_1 және C_{in} кірістерін қосады, S және C_2 береді.
- **OR** — C_1 және C_2 тасымалдарын біріктіріп, C_{out} шығарады.

Жұмыс принципі

1. A және B мәндері HA1-ге беріледі: $S_1 = A \oplus B$, $C_1 = AB$.
2. S_1 және C_{in} мәндері HA2-ге беріледі: $S = S_1 \oplus C_{in}$, $C_2 = S_1 C_{in}$.
3. C_1 және C_2 тасымалдары OR элементінде біріктіріледі: $C_{out} = C_1 + C_2$.

Қорытынды

- Толық қосқышты қарапайым комбинациялық логикалық элементтерден: екі жартылай қосқыш (Half Adder) және бір OR элементінен құрастыруға болады.
- Бұл құрылғы үш бірразрядты санды қосу функциясын орындайды:

$$S = A \oplus B \oplus C_{in}, \quad C_{out} = AB + (A \oplus B) C_{in}.$$

Жартылай қосқыш пен толық қосқышты салыстыру

Сипаттама	Жартылай қосқыш (Half Adder)	Толық қосқыш (Full Adder)
Кірістер саны	2	3
Кіріс сигналдары	A, B	A, B, C_{in}
Шығыстар саны	2	2
Шығыс сигналдары	S, C	S, C_{out}
Орындайтын операция	A + B	A + B + C_{in}
Алдыңғы разрядтан тасымалды қабылдау	Жоқ	Иә
Қосындыны есептеу	Бір XOR	Екі XOR
Тасымалды есептеу	AND	AND және OR
Қосынды формуласы	$S = A \oplus B$	$S = A \oplus B \oplus C_{in}$
Тасымал формуласы	$C = AB$	$C_{out} = AB + C_{in}(A \oplus B)$
Мысал	$1 + 1 = 10_2$	$1 + 1 + 1 = 11_2$
Негізгі қолданылуы	Екі битті қосу	Көпразрядты сандарды қосу

Ең маңыздысы: жартылай қосқыш екі битті қоса алады, бірақ алдыңғы разрядтан келген тасымалды қабылдай алмайды. Ал толық қосқыш екі битпен бірге осы тасымалды да есепке алады.

Сондықтан 4, 8, 16 немесе 32 биттік сандарды қосатын құрылғыларда толық қосқыштар негізгі рөл атқарады.

Көпразрядты қосқыштар

Бірразрядты қосқыштар тек жеке биттерді өңдейді.

Ал нақты цифрлық құрылғыларда көбінесе 4, 8, 16, 32 немесе 64 биттік сандар қолданылады.

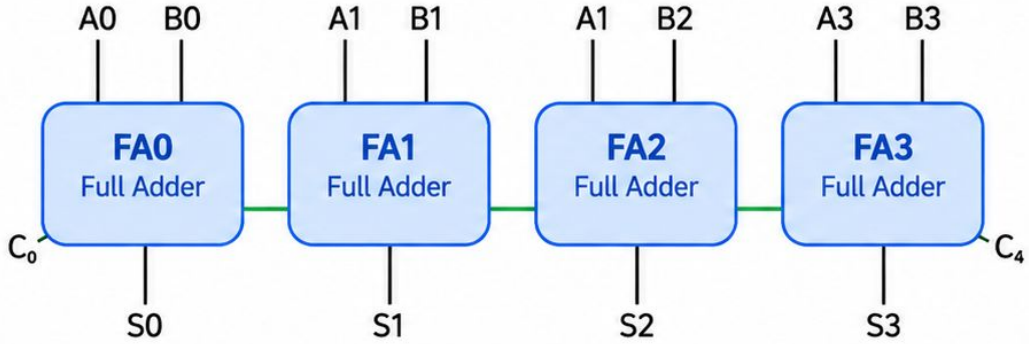
Мұндай сандарды қосу үшін бірнеше толық қосқыш тізбектей біріктіріледі.

Мысалы, төртразрядты қосқыш төрт Full Adder құрылғысынан тұруы мүмкін.

6. Төртразрядты қосқыштың құрылымы

Төртразрядты қосқыш — төрт биттік екілік сандарды қосуға арналған, бірнеше толық қосқыштан (Full Adder) құралған құрылғы.

Құрылымдық схема (4-разрядты қосқыш)



Негізгі түсіндірме

- **FA0–FA3** — төрт разрядқа арналған толық қосқыштар (Full Adder).
- Әр разрядтың тасымалы (carry) келесі разрядқа беріледі:

$C_0 \rightarrow FA0 \rightarrow C_1 \rightarrow FA1 \rightarrow C_2 \rightarrow FA2 \rightarrow C_3 \rightarrow FA3 \rightarrow C_4$

- Суретте тасымалдың логикалық бағыты солдан оңға көрсетілген.

Атауы

Ripple Carry Adder

Мұндай құрылғы тасымалды тізбектей тарататын қосқыш деп аталады.

Жұмыс принципі

1. Ең кіші разрядтағы биттер қосылады.
2. Пайда болған тасымал келесі разрядқа беріледі.
3. Келесі разряд өз биттерімен бірге тасымалды да қосады.
4. Процесс барлық разрядтар бойынша қайталанады.

Қорытынды

- ✔ Төртразрядты қосқыш төрт толық қосқыштан тұрады.
- ✔ Бұл құрылғы көпразрядты екілік сандарды қосу үшін қолданылады, тасымал разрядтар арасында тізбектей («ripple») таралады.

Артықшылығы
Құрылысы қарапайым, логикалық элементтерден оңай жүзеге асырылады.

Кемшілігі
Әр разряд алдыңғы разрядтағы тасымалдың есептелуін күтуі мүмкін.
Сондықтан разрядтар саны артқан сайын есептеу кідірісі ұлғаяды.

Жылдамдықты арттыру үшін тасымалды алдын ала есептейтін құрылымдар, мысалы Carry Lookahead Adder, қолданылады.

7. Екінің толықтаушы коды арқылы азайту

Цифрлық құрылғыларда азайту операциясын да қосқыш арқылы орындауға болады.

Негізгі принцип

- Азайту үшін екінің толықтаушы коды (Two's Complement) қолданылады.

$$A - B = A + \overline{B} + 1$$

Мұндағы $\overline{B}$ — B санының барлық биттерін терістеу нәтижесі.

Мысал: 7 – 3

1. **Төрт биттік екілік сандар:**
   $7 = 0111_2$, $3 = 0011_2$
2. **Бірінші қадам — 3 санының биттерін терістейміз:**
   $0011 \rightarrow 1100$
3. **Екінші қадам — 1 қосамыз:**
   $1100 + 1 = 1101$
4. **Үшінші қадам — 7 санына қосамыз:**
   $0111 + 1101 = 1\ 0100$

Қосу процесі (4 бит)

	C	A (7)	$\overline{B}+1$ (–3)		
	0	0	1	1	1
+		1	1	0	1
	1	0	1	0	0

Сол жақтағы тасымал (1) алынып тасталады.

$0100_2 = 4_{10}$

Қорытынды

- ✓ Төрт биттік нәтиже үшін сол жақтағы соңғы тасымалды алып тастаймыз.
- ✓ Нәтиже дұрыс: $7 - 3 = 4$

Жалпы ескерту

- ✓ Екінің толықтаушы коды арқылы азайту — санды қосу операциясына келтіру әдісі.
- ✓ Осылайша, бір арифметикалық схема қосу және азайту операцияларын орындай алады.

9. Қосу және азайту операцияларын бір құрылғыда орындау

Көптеген процессорларда қосу мен азайту үшін ортақ арифметикалық тізбек қолданылады.

Басқарушы сигнал M

M

- $M = 0$ — қосу $(A + B)$.
- $M = 1$ — азайту $(A - B)$.

Екінші операндты өңдеу

$B' = B \oplus M$

B	M	B'
0	0	0
0	1	1
1	0	1
1	1	0

Бастапқы тасымал

$C_{in} = M$

- $M = 0$: $C_{in} = 0$ (тасымал жоқ).
- $M = 1$: $C_{in} = 1$ (екінің толықтауыш әдісі қолданылады).

Нәтижелік формулалар

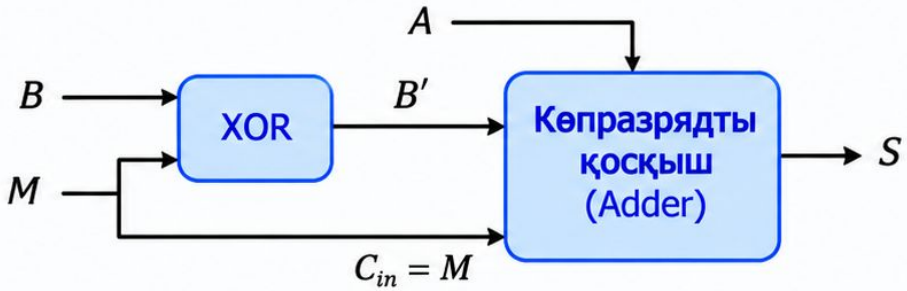
- $M = 0$ (қосу):

$$A + B.$$

- $M = 1$ (азайту):

$$A + \overline{B} + 1 = A - B.$$

Жұмыс схемасы (көпразрядты қосқыш)



Қорытынды

- Бір арифметикалық тізбек арқылы қосу және азайту операциялары орындалады.
- Екінші операнд M сигналына байланысты $B' = B \oplus M$ түрінде өңделеді.
- Бастанқы тасымал $C_{in} = M$ болып алынады.
- Нәтижесінде: $M = 0 \rightarrow A + B$, $M = 1 \rightarrow A - B$.

Арифметикалық-логикалық құрылғы (ALU)

ALU (Arithmetic Logic Unit) — арифметикалық және логикалық операцияларды орындайтын цифрлық құрылғы.

ALU процессордың негізгі есептеу бөліктерінің бірі болып табылады.
Ол келесі операцияларды орындай алады:

- Қосу және азайту.
- AND, OR, XOR логикалық операциялары.
- Сандарды салыстыру.
- Биттерді ығыстыру (құрылғының архитектурасына байланысты).
- Басқа арифметикалық операциялар.

ALU құрылымы

ALU (Arithmetic Logic Unit) — арифметикалық және логикалық операцияларды орындайтын процессордың негізгі бөлігі.

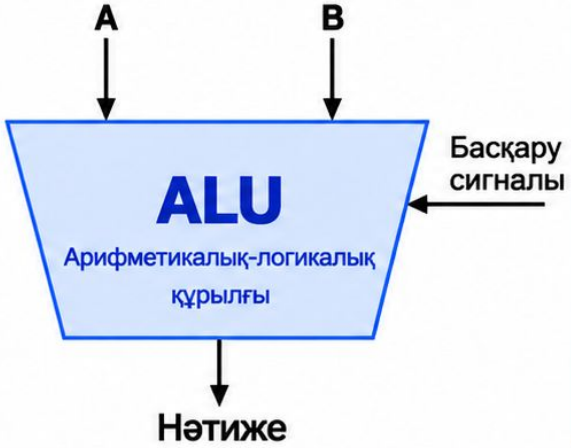
ALU-дың негізгі кірістері

- **A** — бірінші операнд.
- **B** — екінші операнд.
- **Басқару сигналдары** — орындалатын операцияны таңдайды.

ALU-дың шығыстары

- Есептеу нәтижесі (Нәтиже).
- Күй жалаушалары (мысалы, Zero, Carry, Overflow).

ALU-дың құрылымдық схемасы



ALU операцияларының мысалы

Басқару коды	Операция	Нәтиже
000	A + B	Қосу
001	A – B	Азайту
010	A AND B	Логикалық ЖӘНЕ
011	A OR B	Логикалық НЕМЕСЕ
100	A XOR B	Айрықша НЕМЕСЕ

ALU-дың негізгі мүмкіндіктері

- ✓ Арифметикалық операциялар: қосу, азайту және т.б.
- ✓ Логикалық операциялар: AND, OR, XOR және т.б.
- ✓ Нәтижеге байланысты күй жалаушаларын орнату (Zero, Carry, Overflow).

Қорытынды

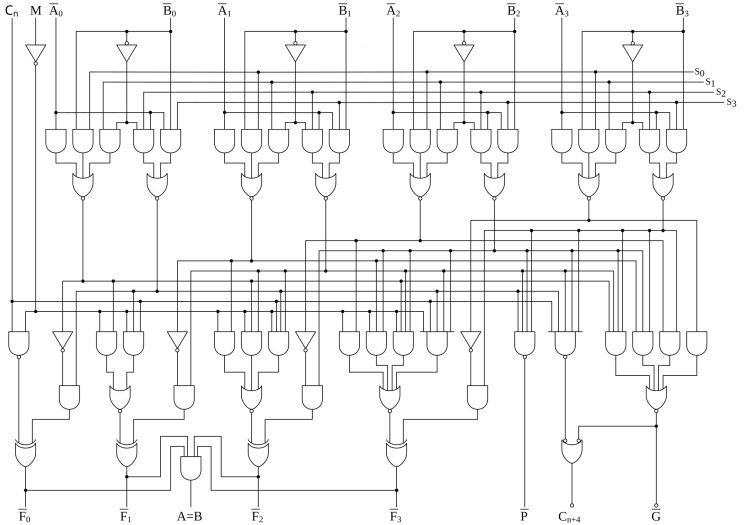
- ✓ ALU — процессорда арифметикалық және логикалық операцияларды орындайтын негізгі құрылғы.
- ✓ Басқару кодтары арқылы орындалатын операциялар таңдалып, нәтиже және күй жалаушалары шығарылады.

ALU міндетті түрде жеке микросхема болуы шарт емес.

Қазіргі заманғы процессорларда ол әдетте процессор кристалының ішінде орналасады.

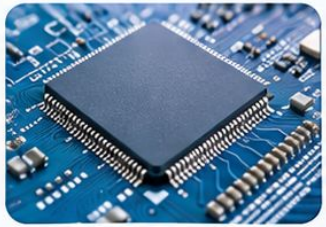
Дегенмен, жеке ALU микросхемалары да бар. Мысалы, 74181 — төртразрядты арифметикалық-логикалық құрылғы.
Бұл — төртразрядты ALU-дың нақты интегралдық микросхемасы. Ол 16 арифметикалық және 16 логикалық операцияны орындай алады.





Қосқыштар мен ALU қолданылатын электрондық жүйелер

1 Микропроцессорлар



- ✓ Арифметикалық есептеулерді орындайды
- ✓ Логикалық операцияларды орындайды
- ✓ Бағдарламалық командаларды жүзеге асырады

CPU → → →

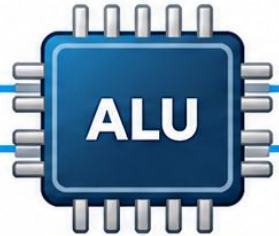
2 Өнеркәсіптік контроллерлер (PLC)



- ✓ Датчиктерден алынған мәндерді өңдейді
- ✓ Қателіктерді есептейді
- ✓ Басқару алгоритмдерін орындайды

Датчиктер → PLC → Басқару (атқарушы құрылғылар)

Электрондық жүйелер



Қосқыштар (Adder)

3 Электрондық есептеу құрылғылары



- ✓ Сандармен арифметикалық операцияларды орындайды
- ✓ Деректерді өңдейді
- ✓ Есептеу тапсырмаларын жүзеге асырады

Есептеу → Деректерді өңдеу → Нәтиже

4 Автомобильдің электрондық басқару жүйелері (ECU)



- ✓ Қозғалтқыш параметрлерін есептейді
- ✓ Аккумулятордың (BMS) күйін бақылайды
- ✓ Басқа жүйелерді (ABS, ESP және т.б.) басқарады

Қозғалтқыш, Аккумулятор (BMS) → ECU → ABS/ESP, Басқа жүйелер

Негізгі идея

Қосқыштар мен ALU деректерді өңдеу, есептеу және басқару міндеттерін орындайды, сондықтан олар көптеген электрондық жүйелердің негізі болып табылады.

Қорытынды

Бұл дәрісте цифрлық құрылғылардағы арифметикалық операциялардың орындалу принциптері қарастырылды.
Негізгі қорытындылар:

- Екілік сандарды қосу логикалық элементтердің көмегімен орындалады.
- Жартылай қосқыш екі битті қосады және қосынды мен тасымалды қалыптастырады.
- Толық қосқыш алдыңғы разрядтан келетін тасымалды да ескереді.
- Көпразрядты қосқыш бірнеше бірразрядты қосқыштан құралады.
- Азайту операциясын екінің толықтауыш кодын пайдаланып, қосу арқылы орындауға болады.
- ALU арифметикалық және логикалық операцияларды біріктіріп орындайды.

Есте сақтайтын негізгі формулалар:

Жартылай қосқыш:

$$\boxed{S = A \oplus B, \qquad C = AB}$$

Толық қосқыш:

$$\boxed{S = A \oplus B \oplus C_{in}}$$

$$\boxed{C_{out} = AB + C_{in}(A \oplus B)}$$

Азайту:

$$\boxed{A - B = A + \overline{B} + 1}$$

Бақылау сұрақтары

1. Цифрлық құрылғыларда арифметикалық операциялар қалай орындалады?
2. Екілік сандарды қосудың негізгі ережелері қандай?
3. Жартылай қосқыш дегеніміз не?
4. Жартылай қосқыштың кірістері мен шығыстарын атаңыз.
5. XOR элементі қосқышта қандай қызмет атқарады?
6. Толық қосқыштың жартылай қосқыштан айырмашылығы қандай?
7. Cin және Cout сигналдары нені білдіреді?
8. Толық қосқышты екі жартылай қосқыш арқылы қалай құруға болады?
9. Көпразрядты қосқыш қалай жұмыс істейді?
10. Ripple Carry Adder құрылғысының кемшілігі қандай?
11. Екінің толықтауыш коды не үшін қолданылады?
12. ALU дегеніміз не?
13. ALU қандай операцияларды орындайды?
14. ALU-дың басқару сигналдары не үшін қажет?
15. Қосқыштар микропроцессорларда қандай қызмет атқарады?