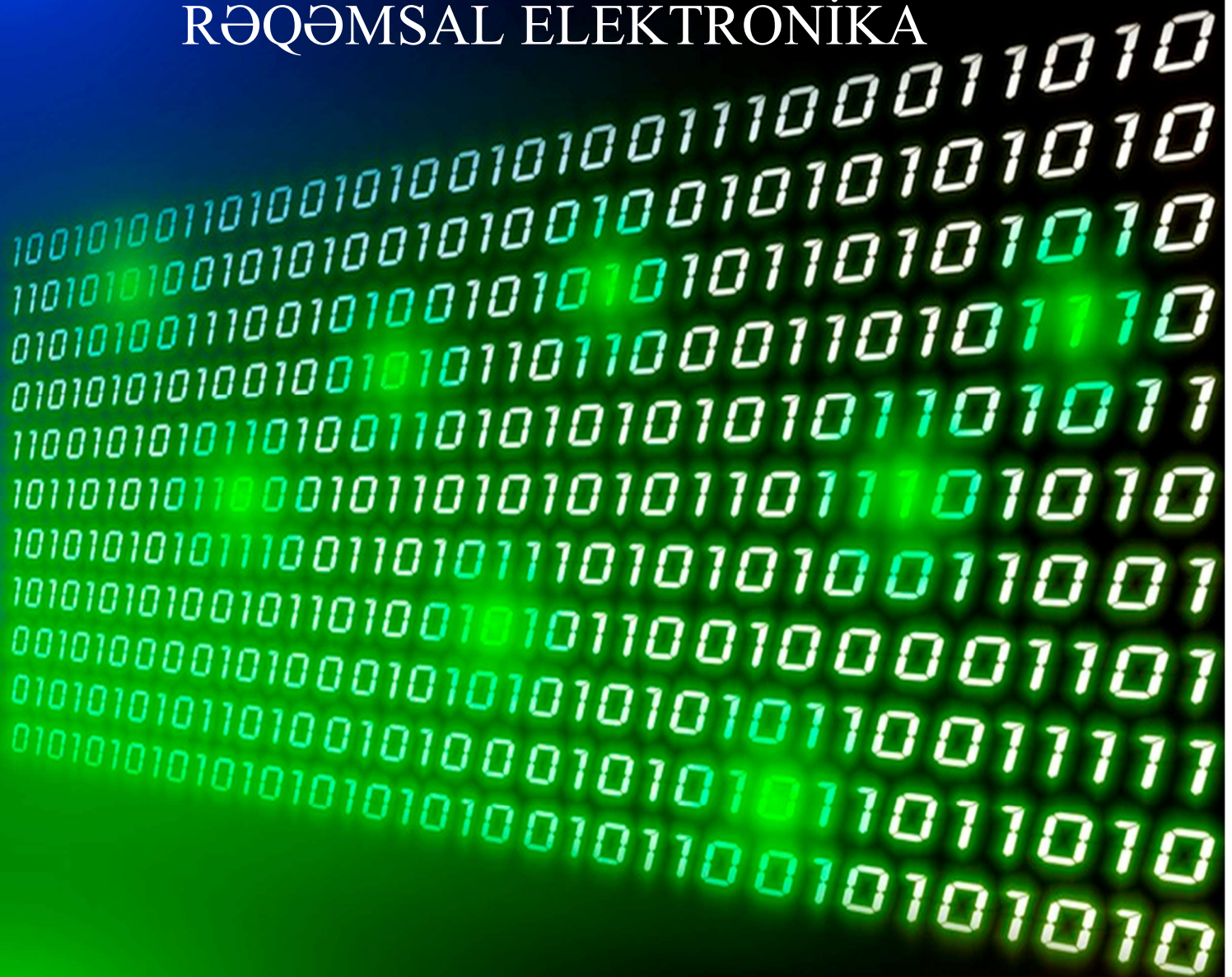


# MİKROPROSESSORLAR



# RƏQƏMSAL ELEKTRONİKA



## **Rəqəmsal elektronika. Giriş.**

Rəqəm metodu və rəqəmsal qurğular müxtəlif inteqrasiyalı inteqrql sxemlərdə tətbiq olunmuş, o cümlədən, mikroprosessor qurğularında, televiziya, radioötürücü, radioqəbuledici və başqa rabitə avadanlıqlarında tətbiq olunur. Buna görə də rəqəmsal sistemləri sahəsində rabitə texniklərinin hazırlığının yüksəldilməsi vacibdir. Belə ki, rəqəmsal qurğularının işlənməsi, təmiri və istismarında bu texniklərin iştirakı tələb olunur.

Bu fənnin əsas məqsədi mürəkkəb rəqəm sistemlərinin öyrədilməsinə əsaslanan impuls, rəqəm və mikroprosessor qurğularının baza hazırlığını formalaşdırmaqdır.

Biz sizinlə mikroprosessor kompleksinə daxil olan mikrosxemlərdən istifadə olunmaqla (etməklə) impuls qurğularının əsas şəbəkələri və rəqəm qurğularının qurulma prinsipi ilə tanış olacağıq.

Rəqəmsal elektronika – EHM və mikroprosessor qurğularının iş prinsipini təşkil edən rəqəm qurğuları ilə təhciz olunmuş məntiqi sxem və məntiqi proqram əsasında yığılmış hissələr arasında əlaqələndirici rolunu oynayır.

Bundan əlavə fənn müasir rəqəmsal qurğuların istifadə olunması ilə EHM-ləri təşkil edən ayrı-ayrı elementlərin və bu elementlər üzərində qurulmuş müxtəlif blok və hissələr arasındakı struktur, funksional, prinsipal əlaqələrin öyrənilməsi ilə bilavasitə məşğul olur.

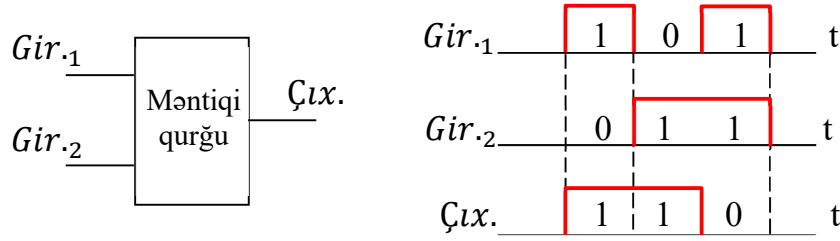
### **Məntiqi funksiya haqqında anlayış.**

Müxtəlif hərəkətləri, əşyaları qeyd etmək üçün sözlərdən istifadə olunur. Sözlər isə əlifba adlanan həriflərdən ibarətdir. Rəqəm texnikasında bu məqsədlə kod sözlərindən istifadə olunur. Bu sözlərin əsas xüsusiyyəti ondan ibarətdir ki, onlar eyni uzunluqda olurlar və onların qurulması üçün cəmi 2 hərifdən istifadə olunur: 0 və 1. beləliklə rəqəmli qurğularda kod sözü müəyyən uzunluqda 0 və 1 simvollarının ardıcılığından ibarətdir. Məsələn: 1011101. Bu sözlərlə rəqəmləri də ifadə etmək olar. Bundan sonra kod sözü haqqında danışdıqda 1 və 0 həriflərini 0 və 1 rəqəmləri ilə qarışıq salmamaq üçün biz onları məntiqi 0 və məntiqi 1 adlandıracağıq. Əgər kod sözünün uzunluğu  $n$  mərtəbədən ibarətdirsə onda kod sözünün  $2^n$  müxtəlif kombinasiyalarını düzəltmək olar. Məsələn,  $n=3$  olduqda  $2^3 = 8$  kod sözünü qurmaq olar: 000, 001, 010, 011, 100, 101, 110, 111.

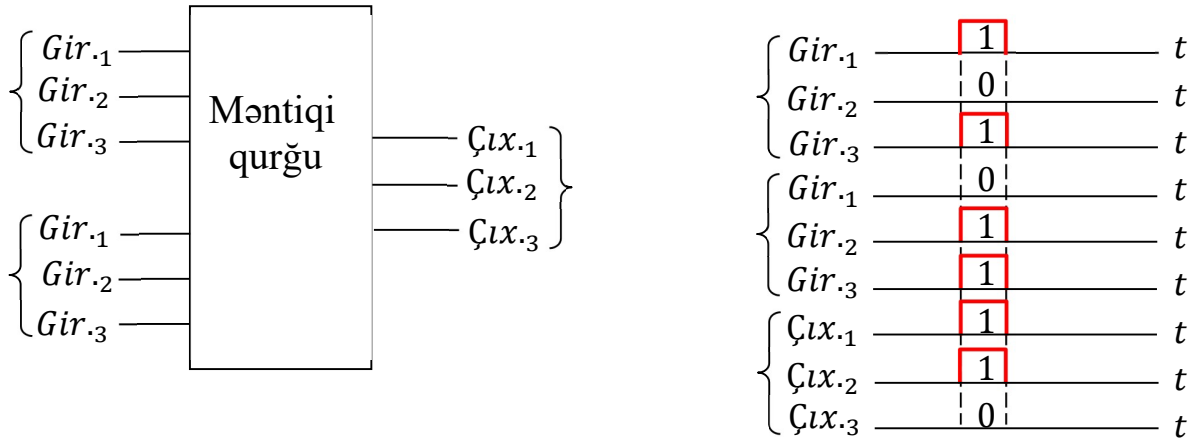
Mürəkkəb rəqəmli qurğuların ayrı-ayrı şəbəkələri arasında ötürülən informasiya kod sözü formasında verilir. Belə ki, hər bir şəbəkənin girişinə kod sözü daxil olur, şəbəkənin çıxışında daxil olan kod sözünün nəticəsinə uyğun olan çıxış kod sözü formalaşır. Çıxış sözü şəbəkənin girişinə daxil olan kod sözündən asılıdır. Ona görə də demək olar ki, çıxış sözü o funksiyadır ki, onun arqumenti giriş sözüdür. Belə funksiyanın əsas xüsusiyyəti odur ki, funksiya və onun arqumenti məntiqi 0 və məntiqi 1 mahiyyətini ala bilər. Bundan sonra belə funksiyaları riyazi məntiqi funksiya adlandıracağıq. Riyazi məntiqi funksiyanı (RMF) formalaşdıran qurğular məntiqi qurğular və ya rəqəmli qurğular adlanır.

Məntiqi qurğular kod sözünün giriş və çıxış üsullarına görə: ardıcıl, paralel və qarışıq olaraq 3 qrupa bölünür.

Ardıcıl hərəkətli qurğuların girişinə kod sözünün simvolları eyni zamanda deyil, simvol ardınca simvol daxil olur. Bu formada da çıxış sözü verilir. Belə qurğunu misal olaraq aşağıdakı kimi göstərmək olar.



Paralel qurğuların girişinə kod sözünün bütün  $n$  simvolları eyni zamanda daxil olur və elə bu formada da çıxış sözü əmələ gəlir.



Şəkildən görüldüyü kimi giriş kod sözü üzərində məntiq əməliyyatları paralel formada həyata keçirilir. Qurğunun girişi 2 qrupa bölünür (I; II). Bu girişlər 3 dərəcəli giriş kod sözünü paralel formada qəbul etmək üçündür. Eləcədə çıxışda paralel formada 3 dərəcəli çıxış kod sözü alınır.

Qarışıq hərəkətli qurğularda giriş və çıxış kod sözləri müxtəlif formada olur. Məsələn; giriş sözü ardıcıl, çıxış kod sözü isə paralel formada olur və əksinə. Qarışıq hərəkətli qurğulardan kod sözünün bir formasından başqa formasına keçmək üçün istifadə olunur.

Məntiqi qurğular funksiyalaşdırma üsullarına görə 2 qrupa bölünür: kombinasional və ardıcıl qurğular.

Kombinasional qurğularda (yaddaşsız avtomat), çıxışdakı hər bir simvol (məntiqi 0 və ya məntiqi 1) həmin anda girişdə olan simvollarla (məntiqi 0 və ya məntiqi 1) müəyyən olunur və bu zaman əvvəlki anda qurğunun girişində hansı simvollar yığımının olmasının fərqi yoxdur. Buna görə də kombinasional qurğuların yaddaşı olmur (onlar qurğunun əvvəlki işi haqqında informasiya saxlamırlar).

Ardıcıl qurğularda (və ya yaddaşlı avtomat) çıxış signalı nəinki həmin anda girişdə olan simvollarla, həm də qurğunun daxili vəziyyətindən və bu vəziyyətin qurğunun əvvəlki zaman anlarında olduğu simvollar yığımından da asılıdır. Buna görə də demək olar ki, ardıcıl qurğular yaddaşa malikdir. Onlar özündə qurğunun əvvəlki vəziyyəti haqqında informasiya saxlayır.

## Elementar məntiqi funksiyalar.

Klassik riyaziyyatda funksiyanın çevrilməsinin 2 üsulundan istifadə olunur: analitik və cədvəl. Elə bu üsulla da funksiya verilə bilər. Cədvəl üsulundan istifadə etdikdə həqiqətlər cədvəli adlanan cədvəl qurulur. Bu cədvəldə arqumentin bütün mümkün olan mənalari (variantları) və bu mənalara uyğun məntiqi funksiyalar verilir. 1 arqumentli funksiyanın həqiqətlər cədvəli aşağıdakı kimidir. (Cədvəl 1)

Arqument	Funksiyalar			
$x$	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	1	1
1	0	1	0	1

1 arqumentli funksiya cədvələ uyğun olaraq aşağıdakı kimi ifadə olunur.

$$\begin{aligned} f_0(x) &= 0; & f_1(x) &= x; \\ f_2(x) &= \bar{x}; & f_3(x) &= 1 \end{aligned}$$

Əgər funksiyanın arqumentlərinin sayı  $n$  ə =dirsə, onda arqumentin müxtəlif mənə birləşmələri  $2^n$ ,  $n$  arqumentli funksiyaların sayı isə  $2^{2^n}$  = olacaq. Belə ki,  $n = 2$  olduqda arqumentin mənalari (variantları) sayı  $2^2 = 4$ , funksiyaların sayı isə  $2^4 = 16$  olacaq. Belə funksiyanın həqiqətlər cədvəli aşağıdakı kimi olacaqdır.

(Cədvəl 2)

Arqument		Funksiyalar															
$x_1$	$x_2$	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$	$f_{11}(x)$	$f_{12}(x)$	$f_{13}(x)$	$f_{14}(x)$	$f_{15}(x)$
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Məntiqi funksiyanın yazılışı analitik üsulla da mümkündür. Adi riyaziyyatda funksiyanın analitik formada verilməsi riyazi ifadə formasında olur. Bu halda funksiyanın arqumentlərini xüsusi riyazi əməliyyatlarla əlaqəndirilir. Buna uyğun olaraq məntiqi funksiyanın analitik formada verilmə üsulu funksiyanın arqumentlər üzərində hansı məntiq əməliyyatlarının və hansı ardıcılıqla yerinə yetirilməsi lazım olduğunu göstərir. Cədvəl 3-də məntiqi ifadənin yazılışı zamanı istifadə olunan məntiq əməliyyatları göstərilmişdir.

Funksiyanın həqiqətlər cədvəli (cədvəl 2) ilə məntiq əməliyyatlarının həqiqətlər cədvəlinin (cədvəl 3) müqayisəsindən aşağıdakıları almaq olar.

(Cədvəl 3)

Məntiqi əməliyyatın işarəsi.		Həqiqətlər cədvəli					Necə oxunur	Əməliyyatın adı
Əsas	Əlavə	$x_1$	0	0	1	1		
		$x_2$	0	1	0	1		
$x_1 \cdot x_2$	$x_1 x_2;$ $x_1 \wedge x_2;$ $x_1 \& x_2$	$x_1 \cdot x_2$	0	0	0	1	$x_1 \vee x_2$	Konyuksiya; məntiqi VƏ; məntiqi hasil.
$x_1 \vee x_2$	$x_1 + x_2$	$x_1 \vee x_2$	0	1	1	1	$x_1$ və ya $x_2$	Dizyuksiya; məntiqi VƏ YA; məntiqi cəm.
$x_1 \rightarrow x_2$	$x_1 \supset x_2$	$x_1 \rightarrow x_2$	1	1	0	1	əgər $x_1$ -dirsə, onda $x_2$ ; $x_1$ $x_2$ -nin implikasiyasıdır	İmplikasiya.
$x_1 \sim x_2$	$x_1 \equiv x_2$ $x_1 \leftrightarrow x_2$	$x_1 \sim x_2$	1	0	0	1	$x_1$ ekvivalentdir $x_2$	Ekvivalentlik; bərabərmənəlilik funksiyası.
$x_1 \oplus x_2$	$x_1 \vee x_2$	$x_1 \oplus x_2$	0	1	1	0	ya $x_1$ , ya da $x_2$ ; $x_1$ , $x_2$ -yə ekvivalent deyil.	2-lik modla görə cəm; bərabərmənəli olmayan funksiya; VƏ YA-ni inkar edir
$x_1 \Delta x_2$	$x_1 \nrightarrow \bar{x}_2$ $\bar{x}_1 \nrightarrow x_2$	$x_1 \Delta x_2$	0	0	1	0	$x_2$ -yə görə $x_1$ qadağandır	Qadağa; implikasiya deyil.
$x_1 \mid x_2$	—	$x_1 \mid x_2$	1	1	1	0	$x_1 \vee x_2$ uyğun deyil	Məntiqi VƏ -YOX; Şifər ştrixi; konyuksiyanın inkarı
$x_1 \downarrow x_2$	—	$x_1 \downarrow x_2$	1	0	0	0	$nə x_1$ , $nə x_2$	Məntiqi VƏ YA-YOX; Pirs oxu; Vebb funksiyası; dizyuksiyanın inkarı
$\bar{x}$	$\neg x$	$x$	0	1			$x_1$ deyil	Məntiqi YOX; məntiqi inkar; inversiya.
		$\bar{x}$	1	0				

$$f_1(x_1, x_2) = x_1 \cdot x_2$$

$$f_7(x_1, x_2) = x_1 \vee x_2$$

$$f_{11}(x_1, x_2) = x_2 \rightarrow x_1$$

$$f_2(x_1, x_2) = x_1 \Delta x_2$$

$$f_8(x_1, x_2) = x_1 \downarrow x_2$$

$$f_{12}(x_1, x_2) = \bar{x}_1$$

$$f_4(x_1, x_2) = x_2 \Delta x_1$$

$$f_9(x_1, x_2) = x_1 \sim x_2$$

$$f_{13}(x_1, x_2) = x_1 \rightarrow x_2$$

$$f_6(x_1, x_2) = x_1 \oplus x_2$$

$$f_{10}(x_1, x_2) = \bar{x}_2$$

$$f_{14}(x_1, x_2) = x_1 \mid x_2$$

Cədvəl 2-dəki qalan  $f_0(x_1, x_2) = 0$ ;  $f_3(x_1, x_2) = x_1$ ;  $f_5(x_1, x_2) = x_2$ ;  $f_{15}(x_1, x_2) = 1$  funksiyaları praktiki olaraq maraqlı deyildir.

Bundan sonra bir və iki arqumentli funksiyaları elementar məntiqi funksiya adlandıracağıq. Funksiyanın məntiqi ifadələri 1 məntiqi əməliyyatdan ibarətdirsə belə funksiyalar elementardır.

## Elementar funksiyanın VƏ, VƏ YA, YOX əməliyyatları ilə verilməsi.

1. Qadağa əməliyyatı.

$$x_1 \Delta x_2 = x_1 \cdot \overline{x_2} \quad (1)$$

Bu və sonrakı bərabərlikləri sübut etmək üçün ifadənin sağ və sol hissələrinin arqumentlərinin mənə mənə yığımlarını ayrı-ayrılıqda uyğunluğunu və həqiqiliyini yoxlayaq.

$x_1$	$x_2$	$x_1 \Delta x_2$
0	0	0
0	1	0
1	0	1
1	1	0

$x_1$	$x_2$	$\overline{x_2}$	$x_1 \cdot \overline{x_2}$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

2. 2-lik modla görə cəm.

$$x_1 \oplus x_2 = x_1 \cdot \overline{x_2} \vee \overline{x_1} \cdot x_2 = (x_1 \vee x_2) \cdot (\overline{x_1} \vee \overline{x_2}) \quad (2)$$

$x_1$	$x_1$	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

$x_1$	$x_1$	$x_1 \cdot \overline{x_2}$	$\overline{x_1} \cdot x_2$	$x_1 \cdot \overline{x_2} \vee \overline{x_1} \cdot x_2$
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

$x_1$	$x_1$	$(x_1 \vee x_2)$	$(\overline{x_1} \vee \overline{x_2})$	$(x_1 \vee x_2) \cdot (\overline{x_1} \vee \overline{x_2})$
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

3. Vəb funksiyası (VƏ YA – YOX əməliyyatı).

$$x_1 \downarrow x_2 = \overline{(x_1 \vee x_2)} \quad (3)$$

$x_1$	$x_2$	$x_1 \downarrow x_2$
0	0	1
0	1	0
1	0	0
1	1	0

$x_1$	$x_2$	$x_1 \vee x_2$	$\overline{(x_1 \vee x_2)}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

4.Məntiqi bərabərmənəlilik.

$$x_1 \sim x_2 = (\overline{x_1 \oplus x_2}) = x_1 \cdot x_2 \vee \overline{x_1} \cdot \overline{x_2} = (\overline{x_1} \vee x_2) \cdot (x_1 \vee \overline{x_2}) \quad (4)$$

Birinci bərabərliyin həqiqiliyi məntiqi bərabərmənəlilik və 2-lik modla görə cəmin həqiqətlər cədvəlinə əsasən təyin olunur. Növbəti bərabərlik – (2) ifadəsinin sağ və sol tərəflərinin inkarından və sağ tərəfin De Mörqan formuluna əsasən çevrilməsindən alınır.

5.İmplikasiya.

$$x_1 \rightarrow x_2 = \overline{x_1} \vee x_2 \quad (5)$$

$x_1$	$x_2$	$x_1 \rightarrow x_2$
0	0	1
0	1	1
1	0	0
1	1	1

$x_1$	$x_2$	$\overline{x_1}$	$\overline{x_1} \vee x_2$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

6.Şifffer funksiyası (VƏ – YOX əməliyyatı)

$$x_1 | x_2 = (\overline{x_1 \cdot x_2}) \quad (6)$$

$x_1$	$x_2$	$x_1   x_2$
0	0	1
0	1	1
1	0	1
1	1	0

$x_1$	$x_2$	$x_1 \cdot x_2$	$(\overline{x_1 \cdot x_2})$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

1. Qadağa əməliyyatı.

$$x_1 \Delta x_2 = x_1 \cdot \bar{x}_2 \quad (1)$$

2. 2-lik modla görə cəm.

$$x_1 \oplus x_2 = x_1 \cdot \bar{x}_2 \vee \bar{x}_1 \cdot x_2 = (x_1 \vee x_2) \cdot (\bar{x}_1 \cdot \bar{x}_2) \quad (2)$$

3. Vebb funksiyası ( $\vee \ominus$  YA – YOX əməliyyatı).

$$x_1 \downarrow x_2 = \overline{x_1 \vee x_2} \quad (3)$$

4. Məntiqi bərabərmənəlilik.

$$\begin{aligned} x_1 \sim x_2 &= \overline{(x_1 \oplus x_2)} = x_1 \cdot x_2 \vee \bar{x}_1 \cdot \bar{x}_2 = \\ &= (\bar{x}_1 \vee x_2) \cdot (x_1 \vee \bar{x}_2) \end{aligned} \quad (4)$$

5. İmplikasiya.

$$x_1 \rightarrow x_2 = \bar{x}_1 \vee x_2 \quad (5)$$

6. Şiffer funksiyası ( $\vee \ominus$  – YOX əməliyyatı)

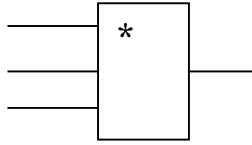
$$x_1 | x_2 = \overline{(x_1 \cdot x_2)} \quad (6)$$



## Məntiqi elementləri haqqında ümumi məlumat.

Məntiqi element:

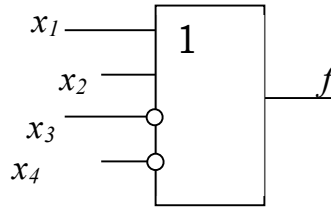
\* - funksiyanın işarəsi.



Bir neçə düz və inkar girişli element.

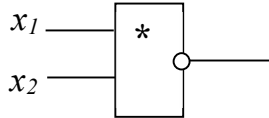
$$f = \varphi(x_1, x_2, x_3, x_4)$$

$$f = x_1 x_2 \vee \bar{x}_3 \bar{x}_4$$

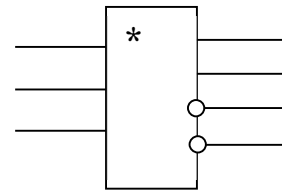


İnkar çıxışlı element.

$$\bar{f} = \overline{x_1 x_2}$$



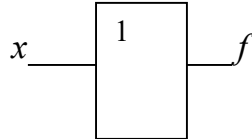
Bir neçə düz və inkar çıxışlı element.



## Məntiqi funksiyaları həyata keçirən elementlər.

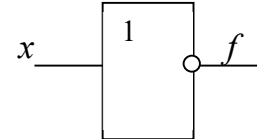
*Təkrarlayıcı*

$$f = x$$



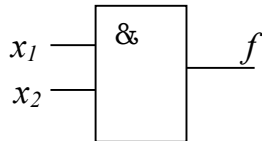
*İnkar (YOX)*

$$f = \bar{x}$$



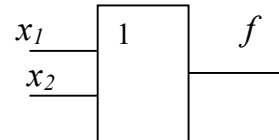
*Konyuktor (VƏ)*

$$f = x_1 \cdot x_2$$



*Dizyuktor (VƏ YA)*

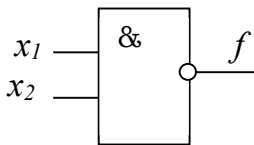
$$f = x_1 \vee x_2$$



*Şiffer elementi.*

*(VƏ - YOX)*

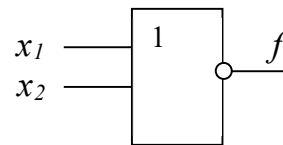
$$f = \overline{(x_1 \cdot x_2)} = x_1 \downarrow x_2$$



*Pirs elementi.*

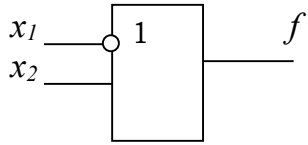
*(VƏ YA - YOX)*

$$f = \overline{(x_1 \vee x_2)} = x_1 \uparrow x_2$$



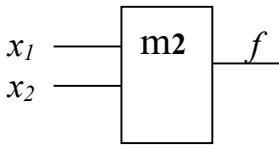
İmplikator.

$$f = \overline{x_1} \vee x_2 = x_1 \rightarrow x_2$$



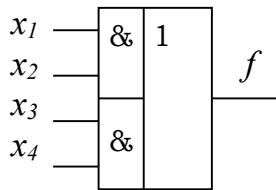
2-lik modla görə cəm.

$$f = x_1 \cdot x_2 \vee \overline{x_1} \cdot \overline{x_2} = x_1 \oplus x_2$$



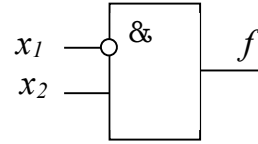
VƏ – VƏ YA

$$f = x_1 \cdot x_2 \vee x_3 \cdot x_4$$



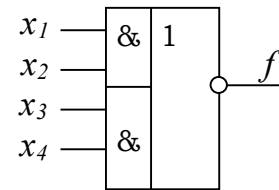
Qadağa

$$f = \overline{x_1} \cdot x_2 = x_2 \Delta x_1$$



VƏ-VƏ YA-YOX

$$f = \overline{(x_1 \cdot x_2 \vee x_3 \cdot x_4)}$$



## Triggerlər. Təyinatı, növləri.

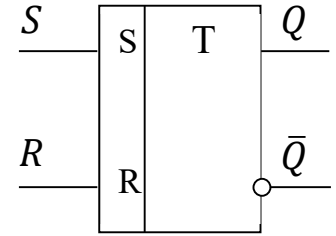
Rəqəm sistemlərində məntiq elementlərindən başqa, həm də 2-lik məlumatların yadda saxlanması üçün yaddaş elementlərindən də istifadə olunur. Məlumatı yadda saxlama prinsipinə görə yaddaş elementləri 2 növə bölünür:

- 1) Statiki yaddaş elementləri.
- 2) Dinamiki yaddaş elementləri.

İstərsə statiki, istərsə də dinamiki YE+nin 2 dayanıqlıq vəziyyəti olur. Onlardan biri 0-a, digəri 1-ə uyğun gəlir. Ən sadə halda EHM+də YE kimi triggerlərdən istifadə olunur.

2 dayanıqlıq vəziyyətindən birində ola bilən və giriş signalının təsiri ilə bir vəziyyətdən digərinə keçə bilən qurğulara triggerlər deyilir.

Uyğun olaraq triggerin dayanıqlıq vəziyyəti kimi 1 və 0 götürülür və onun vəziyyəti çıxış signalına görə müəyyənləşir.



Triggerin 2 çıxışı olur: düz  $Q$  və invers  $\bar{Q}$

Triggerin vəziyyəti bu çıxışlardakı gərginlik səviyyəsinə görə müəyyən edilir: əgər  $Q$  çıxışında gərginlik məntiqi 0 səviyyəsinə ( $Q = 0$ ) uyğundursa onda trigger 0 vəziyyətində,  $Q = 1$  olduqda isə trigger məntiqi 1 səviyyəsində olur.

Triggerin müxtəlif növ girişləri olur. Onların təyinatına baxaq:

R – 0 vəziyyətinə gətirmədə bölücü giriş;

S – 1 vəziyyətinə gətirmədə bölücü giriş;

K – universal triggerin 0 vəziyyətinə gətirmə girişi;

J – universal triggerin 1 vəziyyətinə gətirmə girişi;

T – sayma girişi.

D – triggeri girişdəki məntiqi səviyyəyə uyğun səviyyəyə gətirən informasiya girişi;

C – sinxronlaşdırma (idarəedici) girişi.

Triggerlərin adı onların girişinin növünə görə təyin edilir. Məsələn, RS-triggeri – R və S girişləri olan trigger.

Triggerin növləri: Ən çox istifadə olunan triggerlərin ümumi xarakteristikalarına baxaq.

Hər bir triggeri kecid cədvəli xarakterizə edir.

$S$	$R$	$Q$
0	0	$Q_0$
0	1	0
1	0	1
1	1	*

**RS** triggerinin kecid cədvəli aşağıdakı kimidir.

RS triggeri keçid cədvəlinə uyğun olaraq aşağıdakı kimi işləyir. Burada  $Q_0$  – girişə aktiv siqnal veriləndə triggerin olduğu vəziyyətidir.  $R$  və  $S$  girişlərinə aktiv səviyyə verilməyəndə kimi trigger  $Q_0$  vəziyyətində qalır.  $R = 1$  aktiv siqnalı triggeri 0 vəziyyətinə,  $S = 1$  aktiv siqnalı isə triggeri 1 vəziyyətinə gətirir. Cədvəldəki \* işarəsi ilə qeyd olunmuş vəziyyət qadağan olunmuş

rejimdir, çünki bu zaman triggerin hansı səviyyədə olduğunu müəyyən etmək qeyri mümkün olur.

$J$	$K$	$Q$
0	0	$Q_0$
0	1	0
1	0	1
1	1	$\bar{Q}_0$

**JK** triggerinin kecid cədvəli aşağıdakı kimidir.

Bu triggerlər RS triggerlərindən yalnız giriş siqnallarının qadağan olunmuş rejimi ilə fərqlənir. Bu triggerlərdə qadağan olunmuş rejim yoxdur.  $J = K = 1$  olduqda trigger hal hazırda olduğu  $Q_0$  vəziyyətinin inversi vəziyyətinə keçir ( $\bar{Q}_0$ ).

$D$	$Q$
0	0
1	1

**D** triggerinin kecid cədvəlinə baxaq. Həmişə trigger  $D$  girişindəki səviyyəyə uyğun vəziyyətə keçir.

**T** triggerinin kecid cədvəli aşağıdakı kimidir. Girişə  $T=0$  siqnalı verildikdə trigger vəziyyətini dəyişməmiş, yəni siqnal verilməzdən əvvəlki  $Q_0$  vəziyyətində qalır.  $T=1$  siqnalı verildikdə trigger əvvəlki vəziyyətinin inversi vəziyyətinə ( $\bar{Q}_0$ ) keçir.

$T$	$Q$
0	$Q_0$
1	$\bar{Q}_0$

Triggerlərin qurulmasında çıxışdan girişə əks əlaqə əsas rol oynayır və bundan istifadə edib triggerlərin müxtəlif növlərini almaq olar. Funksional imkanlarına görə triggerlər fərqlənir. Triggerlər **sürətliliyi, həssashlığı, maneəyə dayanıqlılığı və funksional imkanları** ilə xarakterizə olunur.

**İti sürətlilik** – triggerin vəziyyətinin qoşulmasının maksimal tezliyi ilə müəyyən olunur və 100-lərlə MqHs olur.

**Həssashlığı** – girişdə ən az gərginlik olduqda belə qoşulmanın baş verməsilə müəyyən olunur.

**Maneəyə dayanıqlılıq** – xəta şəraitində triggerin normal işləmə qabiliyyətidir.

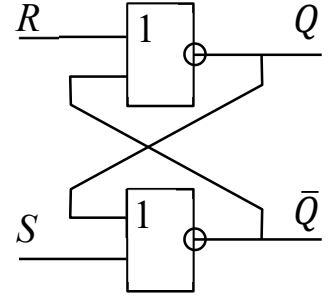
**Funksional imkan** – triggerin funksional imkanı dedikdə onun giriş siqnallarının sayı ilə xarakterizə olunması başa düşülür.

## Asinxron triggerlər.

İnformasiyanın qəbul olunması üsullarına görə triggerlər 2 növə bölünür:

- 1) **Asinxron triggerlər.**
- 2) **Sinxron triggerlər.**

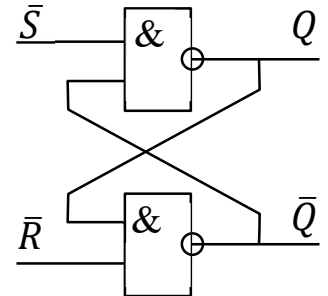
**Asinxron triggerlər.** Qeyd etdik ki, triggerlər əsasən “VƏ YOX”, “VƏ YA-YOX” məntiq elementləri üzərində qurulur. VƏ YA-YOX məntiq elementi üzərində qurulmuş asinxron RS triggerin məntiqi struktur sxemi aşağıdakı kimidir. Trigger iki VƏ YA –YOX məntiq elementi üzərində elə qurulub ki, hər bir məntiq elementinin çıxışı digər məntiq elementinin girişinə qoşulub. Elementlərin belə qoşulması triggerin iki dayanıqlıq vəziyyətindən birində olmasını təmin edir.



Asinxron YE+nin yalnız 2 ədəd informasiya girişi ( 0 və 1-ə uyğun) və uyğun olaraq 2 ədəd çıxışı var. 1- ə uyğun çıxış triggerlərdə düz çıxış kimi nəzərdə tutulur və Q ilə işarə olunur. 0-a uyğun çıxış isə inkar çıxış kimi nəzərdə tutulur və  $\bar{Q}$  ilə işarə olunur.

Əgər trigger “0” vəziyyətindədirsə , bu o deməkdir ki, triggerin inf-ya girişinə (yəni R-ə)  $R = 1$  verilib və inkar çıxışda isə  $\bar{Q} = 1$  alırıq. Bu halda  $S = 0$ ,  $Q = 0$ . Əgər yenə də  $R = 1$  versək, trigger öz vəziyyətini dəyişməyəcək və  $\bar{Q} = 1$ ,  $Q = 0$  olacaq. Əgər S inf-ya girişinə  $S = 1$  versək, onda uyğun olaraq  $Q = 1$ ,  $\bar{Q} = 0$  olacaq. Yəni trigger öz vəziyyətini dəyişir, “0”-dan “1”-ə keçir. Yenə də  $S = 1$  versək, onda yenə də trigger öz vəziyyətini saxlayır ( $Q = 1$ ,  $\bar{Q} = 0$ ). Triggerin vəziyyətini “0” vəziyyətinə uyğun vəziyyətə gətirmək üçün onda R inf-ya girişinə  $R = 1$  veririk və bu halda  $\bar{Q} = 1$ ,  $Q = 0$  alırıq.

Trigger sxemlərində inf-ya girişinin hər 2-nə  $R = 1$ ,  $S = 1$  kombinasiyası qadağan olunmuş kombinasiya kimi nəzərdə tutulur və heç vaxt istifadə olunmur. Ona görə ki, bu girişlərə uyğun suqnalın təsir müddəti qurtqrdıqdan sonra triggerin vahid və ya “0” vəziyyətdə olduğunu müəyyən etmək qeyri mümkün olur.



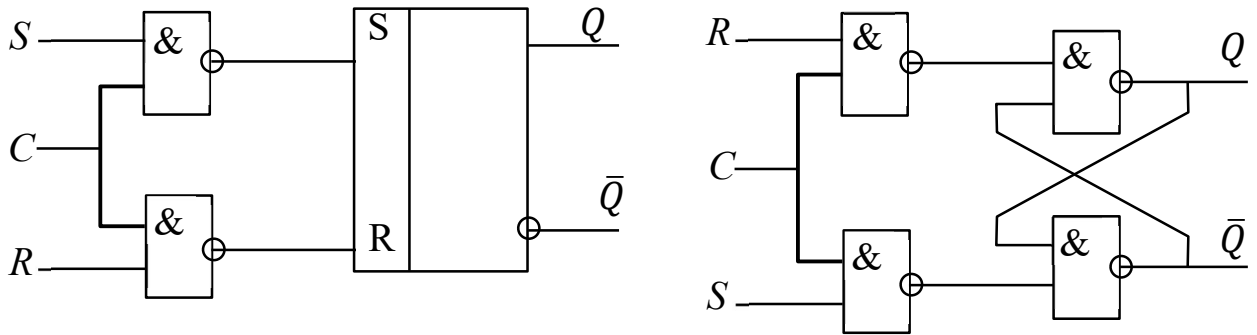
Asinxron RS triggerini VƏ YOX məntiq elementi üzərində də yığmaq mümkündür. Bu halda sxem aşağıdakı kimi olur.

Tutaq ki, ilk vəziyyətdə trigger “0” vəziyyətindədir. Yəni  $Q = 0$ ,  $\bar{Q} = 1$ . Bu o deməkdir ki, triggerin  $\bar{S} = 1$ ,  $\bar{R} = 0$  verilib. Triggerin vəziyyətini dəyişmək

lazım gələrsə, onda  $\bar{S} = 0, \bar{R} = 1$  vermək lazımdır. Bu halda,  $Q = 1, \bar{Q} = 0$ , olacaq. Göstərilən giriş inf+nı1 daha təkrar etsək, yəni  $\bar{S} = 0, \bar{R} = 1$  onda trigger öz vəziyyətini dəyişməyəcək və ilk vəziyyətdə olduğu kimi qalacaq.

### Sinxron triggerlər.

Asinxron triggerlərdən (RS) fərqli olaraq sinxron triggerlərdə informasiya girişi ilə bərabər həm də sinxro girişir olur və triggerin vəziyyəti onun informasiya girişinə verilən siqnalla yanaşı həm də sinxrosiqnaldan da asılı olur. Yəni əgər sinxron triggerin girişlərindən hər hansı birinə 1-ə uyğun siqnal verilərsə və sinxrogirişə heç bir siqnal verilməzsə, onda trigger öz vəziyyətini dəyişmir və əvvəlki vəziyyətində qalmalıdır. Bunun üçün də sinxro siqnalların triggerin girişinə verilməsi üçün əlavə məntiq elementləri sxemə qoşulmalıdır. Bu halda VƏ YOX məntiq elementi üzərində yığılmış sinxron triggerin (RS) sxemi aşağıdakı kimi olcaqdır.



Sinxron RS triggerində əgər inf-ya girişlərindən 1-nə “1”-ə uyğun siqnal verilərsə, lakin onun sinxro girişinə sinxro impuls verilməzsə, onda trigger öz vəziyyətini dəyişmir. Yəni əvvəlki vəziyyətində qalacaq. Bu halda  $S = 1, C = 0; R = 1, C = 0$  halına uyğun gəlir. Əgər triggerin vəziyyətini “1” vəziyyətinə gətirmək istəsək onun girişinə  $S = 1$  vermək lazımdır. Sinxro girişə isə müəyyən intervalla dəyişən sinxro takt siqnalları vermək lazımdır. Əgər triggerə yenə də  $R = 1, C = 0, S = 0$  versək trigger vəziyyətini dəyişmir. Əgər  $R = 1, C = 1, S = 0$  olarsa triggerin çıxış vəziyyəti  $Q = 0, \bar{Q} = 1$  olacaq.

## Şifratorlar.

Şifrator 10-luq rəqəmi 2-lik say sisteminə keçirən qurğudur. Tutaq ki, şif-un 10-luq rəqəmlə qeyd olunmuş  $(0,1,2,3,4,...m-1)$  m girişi və n çıxışı vardır. Girişlərdən birinə siqnalın verilməsi çıxışda təsirlənmiş girişə uyğun n dərəcəli 2-lik rəqəmin əmələ gəlməsinə gətirib çıxarır. Aşağıdakı şəkildə 10-luq  $0,1,2,3,4,...,9$  rəqəmin 8421 kodunda 2-lik formaya çevirən şifrator simvolik olaraq göstəril-mişdir.

CD- ingilis sözü olan CODER sözündən əmələ gələn həriflərdir. Cədvəldə 10-luq və 2-lik kodların uyğunluğundan görünür ki, 1 çıxış şınında dəyişən  $x_1$  məntiqi “1” səviyyəsində o vaxt olur ki, dəyişən  $y_1, y_3, y_5, y_7, y_9$  girişlərindən biri bu səviyyədə olsun. Göründüyü kimi:

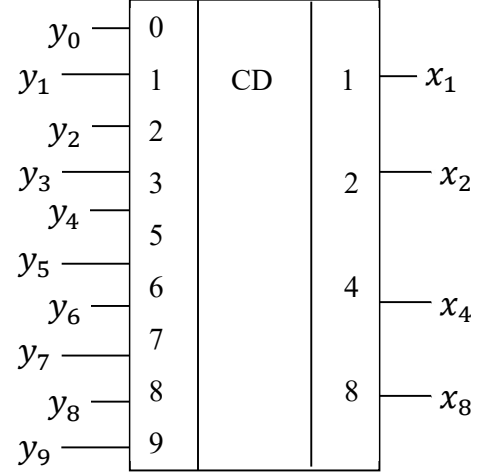
$$x_1 = y_1 \vee y_3 \vee y_5 \vee y_7 \vee y_9$$

Digər çıxışlar üçün:

$$x_2 = y_2 \vee y_3 \vee y_6 \vee y_7$$

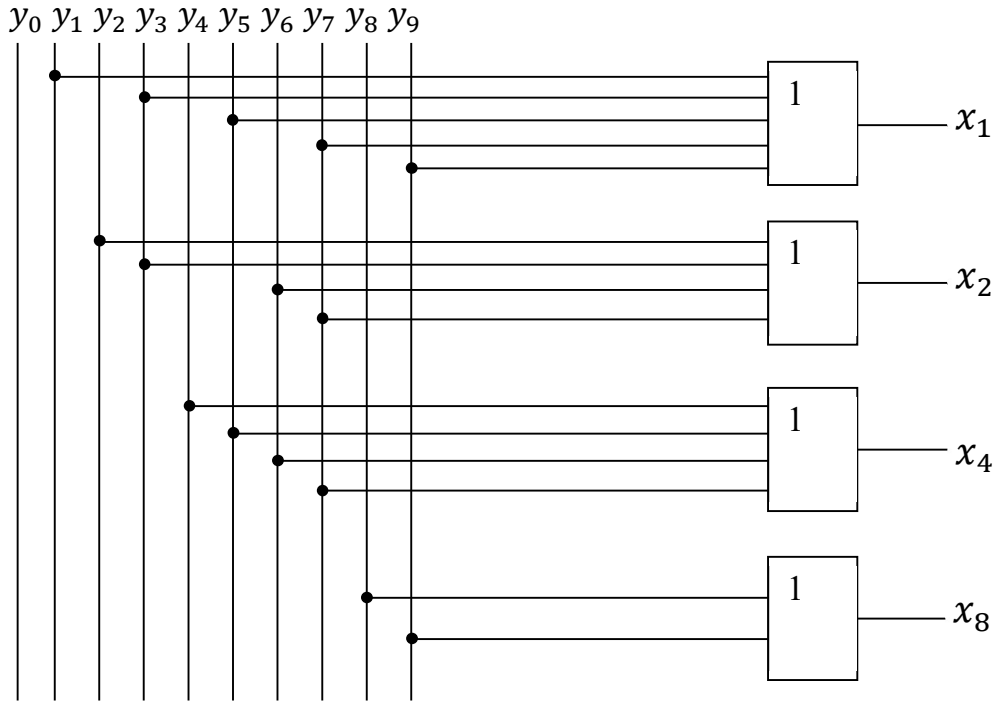
$$x_4 = y_4 \vee y_5 \vee y_6 \vee y_7 \quad (1)$$

$$x_8 = y_8 \vee y_9$$



Onluq rəqəm	İkilik kod			
	$x_8$	$x_4$	$x_2$	$x_1$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Alınan (1) ifadəsinə uyğun sxem aşağıdakı kimidir.



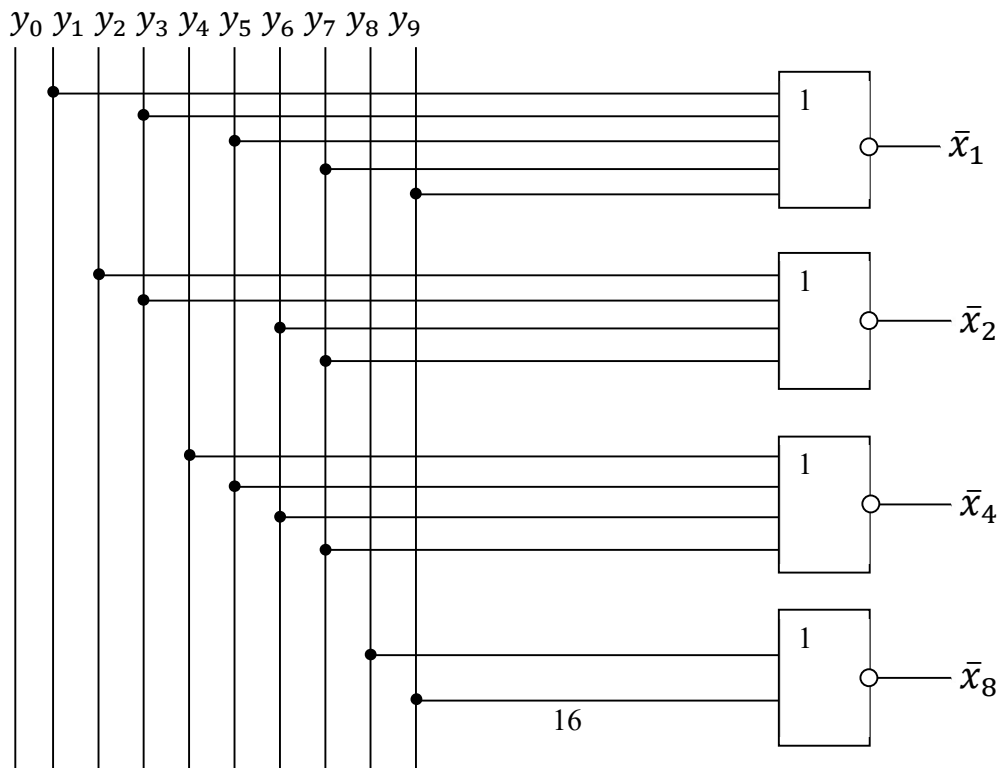
Aşağıdaki şəkildə VƏ YA-YOX elementi üzərində qurulmuş şifratorun sxemi verilmişdir. Şifrator (2) ifadəsinə uyğun qurulmuşdur.

$$\overline{x_1} = \overline{y_1 \vee y_3 \vee y_5 \vee y_7 \vee y_9} = y_1 \downarrow y_3 \downarrow y_5 \downarrow y_7 \downarrow y_9$$

$$\overline{x_2} = \overline{y_2 \vee y_3 \vee y_6 \vee y_7} = y_2 \downarrow y_3 \downarrow y_6 \downarrow y_7$$

$$\overline{x_4} = \overline{y_4 \vee y_5 \vee y_6 \vee y_7} = y_4 \downarrow y_5 \downarrow y_6 \downarrow y_7$$

$$\overline{x_8} = \overline{y_8 \vee y_9} = y_8 \downarrow y_9$$





Bu vaxt şifratorun invers çıxışları olur.

VƏ YOX elementi üzərində şifrator yığıldıqda aşağıdakı məntiqi ifadə sistemindən istifadə etmək olar.

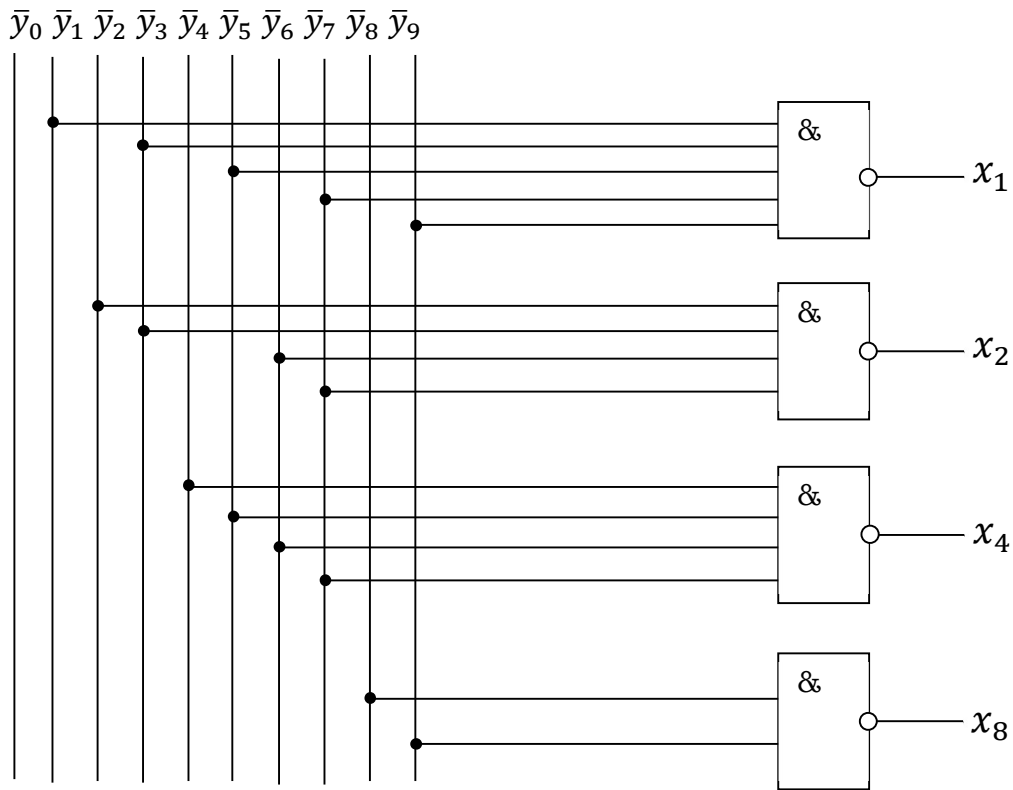
$$x_1 = \overline{y_1 \vee y_3 \vee y_5 \vee y_7 \vee y_9} = \overline{y_1 \cdot y_3 \cdot y_5 \cdot y_7 \cdot y_9} = \overline{y_1} \cdot \overline{y_3} \cdot \overline{y_5} \cdot \overline{y_7} \cdot \overline{y_9}$$

$$x_2 = \overline{y_2} \cdot \overline{y_3} \cdot \overline{y_6} \cdot \overline{y_7}$$

$$x_4 = \overline{y_4} \cdot \overline{y_5} \cdot \overline{y_6} \cdot \overline{y_7}$$

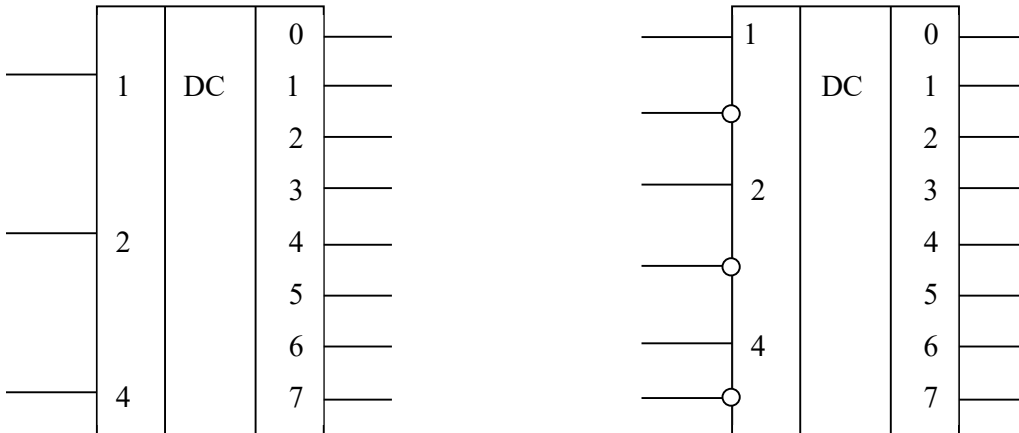
$$x_8 = \overline{y_8} \cdot \overline{y_9}$$

VƏ YOX elementi üzərində qurulmuş şifratorun sxemi belə olacaqdır.



## Deşifratorlar.

2-lik rəqəmi çox da böyük olmayan 10-luq rəqəmə çevirmək üçün deşifratorlardan (decoder) istifadə olunur. Deşifratorun girişləri 2-lik rəqəmləri vermək üçündür. Çıxışları isə ardıcıl olaraq 10-luq rəqəmlərlə nömrələnir. Girişə 2-lik rəqəm verildikdə uyğun çıxışda – (nömrəsi giriş rəqəminə uyğun olan) siq-nal əmələ gəlir. Deşifratorlar çox yerlərdə öz tətbiqini tapmışdır. Ən çox onlardan rəqəmli qurğulardan rəqəm və ya mətnin kağıza çap olunmasında istifadə olunan qurğularda istifadə olunur. Belə qurğularda deşifratorun girişinə daxil olan 2-lik rəqəm onun uyğun çıxışında siq-nalın əmələ gəlməsinə səbəb olur. Bu siq-nalın köməyiylə girişdəki 2-lik rəqəmə uyğun simvollar çap olunur. Aşağıdakı şəkildə deşifratorun simvolik olaraq şərti sxemi göstərilmişdir. DC- ingilis sözü olan DECODER sözünün hərflərindən əmələ gəlmişdir. Deşifratorlar həm də cütfaz (parafaz) girişlərə də malik ola bilər. Bu girişlərlə giriş dəyişənləri ilə yanaşı onların inversləri də verilə bilər.



Qurulma metoduna görə xətti və düzbucaqlı deşifratorları fərqləndirirlər.

**Xətti deşifrator:** aşağıdakı cədvəldə çevrilməni həyata keçirən deşifratorun qurulmasına baxaq.

Çıxış dəyişənlərinin qiyməti aşağıdakı məntiqi ifadə ilə təyin edilir.

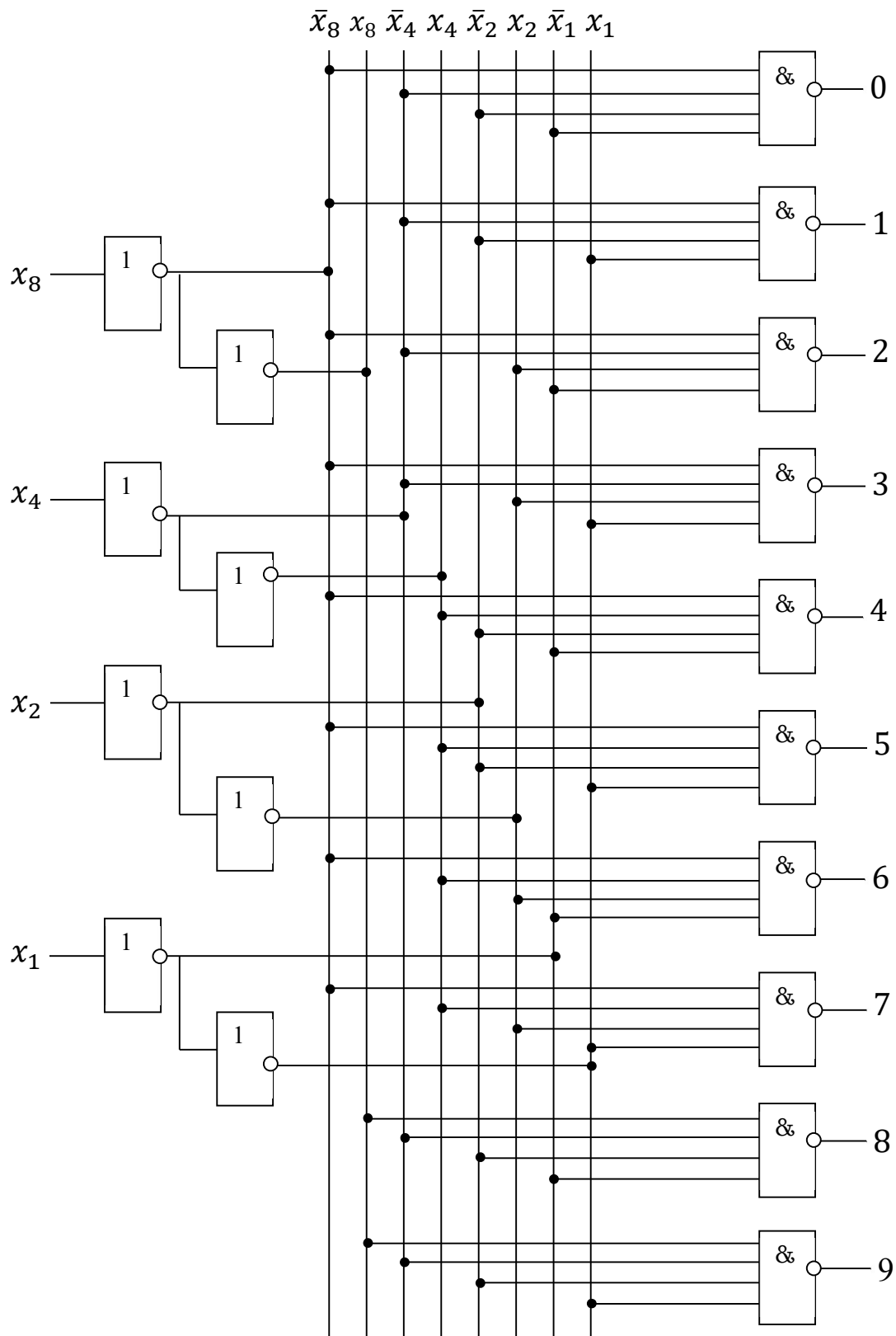
$$\begin{aligned}
y_0 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} \\
y_1 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} \\
y_2 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} \\
y_3 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} \\
y_4 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} \\
y_5 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} \\
y_6 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} \\
y_7 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} \\
y_8 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} \\
y_9 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1}
\end{aligned} \tag{1}$$

8421 giriş kodu				Çıxış №
$x_8$	$x_4$	$x_2$	$x_1$	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

$$\begin{aligned}
y_0 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} = \overline{x_8} \mid \overline{x_4} \mid \overline{x_2} \mid \overline{x_1} \\
y_1 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} = \overline{x_8} \mid \overline{x_4} \mid \overline{x_2} \mid x \\
y_2 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} = \overline{x_8} \mid \overline{x_4} \mid x_2 \mid \overline{x_1} \\
y_3 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} = \overline{x_8} \mid \overline{x_4} \mid x_2 \mid x_1 \\
y_4 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} = \overline{x_8} \mid x_4 \mid \overline{x_2} \mid \overline{x_1} \\
y_5 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} = \overline{x_8} \mid x_4 \mid \overline{x_2} \mid x_1 \\
y_6 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} = \overline{x_8} \mid x_4 \mid x_2 \mid \overline{x_1} \\
y_7 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} = \overline{x_8} \mid x_4 \mid x_2 \mid x_1 \\
y_8 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} = x_8 \mid \overline{x_4} \mid \overline{x_2} \mid \overline{x_1} \\
y_9 &= \overline{x_8 \cdot x_4 \cdot x_2 \cdot x_1} = x_8 \mid \overline{x_4} \mid \overline{x_2} \mid x_1
\end{aligned} \tag{2}$$

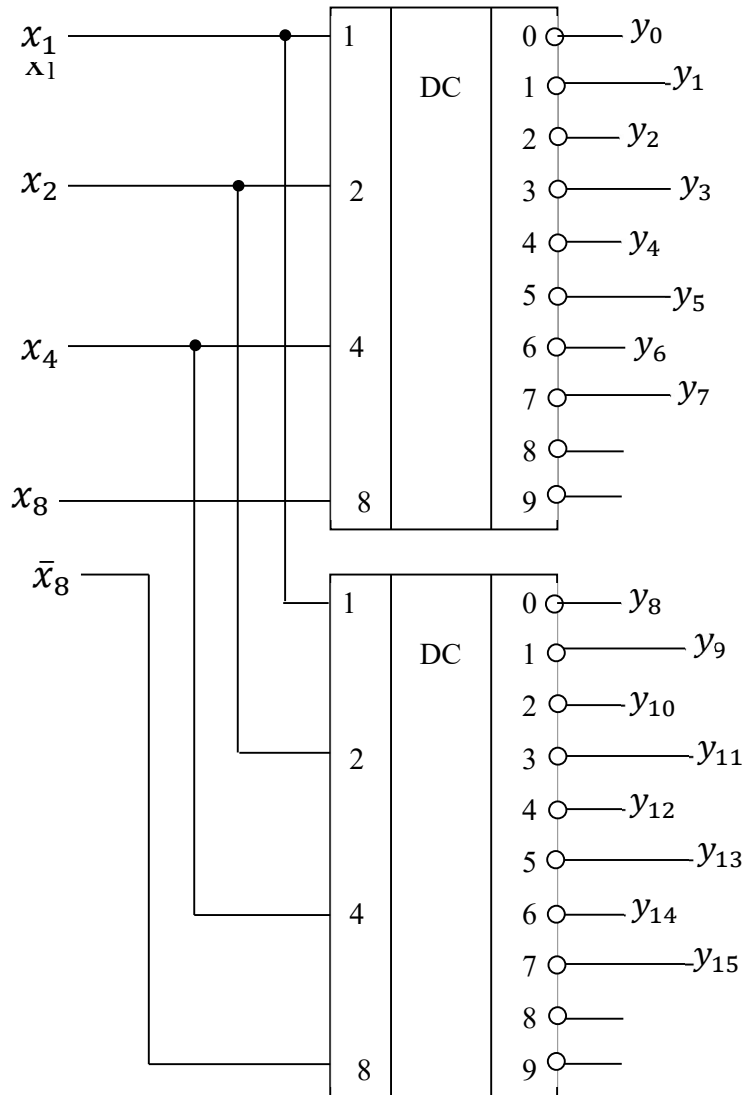
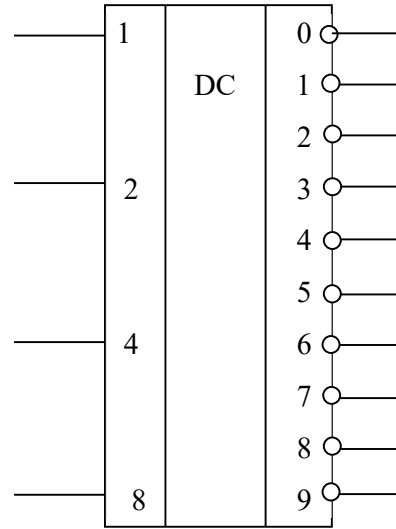
Xətti deşifratorda çıxış dəyişənləri ya (1) ya da (2) ifadəsinə əsasən formalaşır. Deşifratör VƏ YOX elementi üzərində qurulursa onda çıxış funksiyasının inversiyası alınmaqla (2) ifadəsindən istifadə olunur.

Aşağıdakı sxemdə VƏ YOX elementi üzərində qurulmuş deşifratörün struktur sxemi göstərilmişdir.



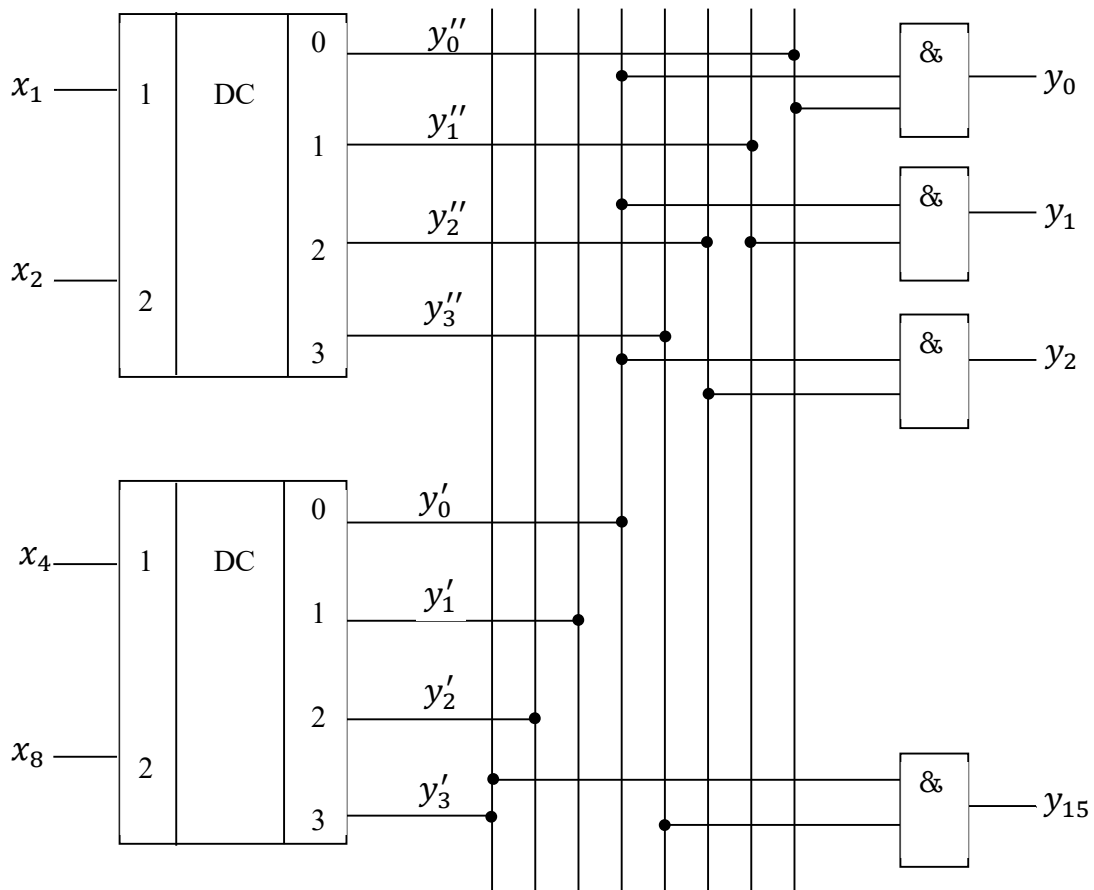
Bu əməliyyatı yerinə yetirən deşifratorun şərti sxemi aşağıdakı kimidir.

Dörd dərəcəli 2-lik 8421 kodunun bütün mümkün kombinasiyalarını deşifrasiya etmək üçün istifadə olunan 16 çıxışlı deşifratoru 10 çıxışlı 2 deşifratorndan istifadə etməklə göstərmək olar. Hər deşifratorun 8 çıxışından istifadə olunur ki, bunlar da lazım olan 16 çıxış əmələ gətirir ( $y_0, y_1, y_2, \dots, y_{15}$ ). Belə deşifratorun struktur sxemi aşağıdakı kimidir.



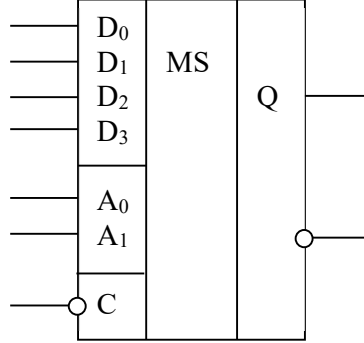
**Düzbucaqlı deşifratorlar.** Düzbucaqlı deşifratorun qurulma prinsipinə 4 girişi və 16 çıxışı olan deşifratorun nümunəsində baxaq. Giriş dəyişənlərini  $x_8, x_4, x_2, x_1$ -i hər birində 2 dəyişən olmaqla 2 qrupa ayıraq ( $x_8, x_4$  və  $x_2, x_1$ ). Hər bir cüt dəyişəni 4 çıxışlı 2 xətti deşifratorun giriş dəyişənləri kimi istifadə edirik. Sxem aşağıdakı kimi olacaqdır. Düzbucaqlı deşifratorun çıxış dəyişənləri aşağıdakı məntiqi ifadə ilə təyin olunur.

$$\begin{aligned}
 y'_0 &= \overline{x_8} \cdot \overline{x_4} & y''_0 &= \overline{x_2} \cdot \overline{x_1} \\
 y'_1 &= \overline{x_8} \cdot x_4 & y''_1 &= \overline{x_2} \cdot x_1 \\
 y'_2 &= x_8 \cdot \overline{x_4} & y''_2 &= x_2 \cdot \overline{x_1} \\
 y'_3 &= x_8 \cdot x_4 & y''_3 &= x_2 \cdot x_1
 \end{aligned}$$



## Multipleksor.

Multipleksor bir neçə girişdən birini seçərək onu öz çıxışına birləşdirən qurğudur. Multipleksorun bir neçə informasiya ( $D_0, D_1, \dots$ ) girişləri, ünvan girişləri ( $A_0, A_1, \dots$ ) girişləri və eləcə də stroblaşdırma (icazə) siqnalı vermək üçün C girişə və bir Q çıxışa malikdir. Aşağıdakı şəkildə 4 informasiya girişi olan multipleksorun simvolik işarəsi verilmişdir.



Multipleksorun hər bir informasiya girişinə ünvanın verdiyi ünvan verilir. C girişinə stroblaşdırma siqnalı verildikdə multipleksor ünvanı 2-lik kodla, ünvan girişindən verilən informasiya girişlərindən birini seçir və onu çıxışa qoşur. Belə-liklə də, müxtəlif informasiya girişlərinin ünvanını ünvan girişlərindən verməklə, bu girişlərdəki rəqəmli siqnalları Q çıxışına ötürmək olar. Multipleksorun funksiyalaşdırılması aşağıdakı cədvəllə təyin edilir.

Ünvan girişləri		Strob siqnalı	Çıxış
$A_1$	$A_0$	C	Q
X	X	0	0
0	0	1	$D_0$
0	1	1	$D_1$
1	0	1	$D_2$
1	1	1	$D_3$

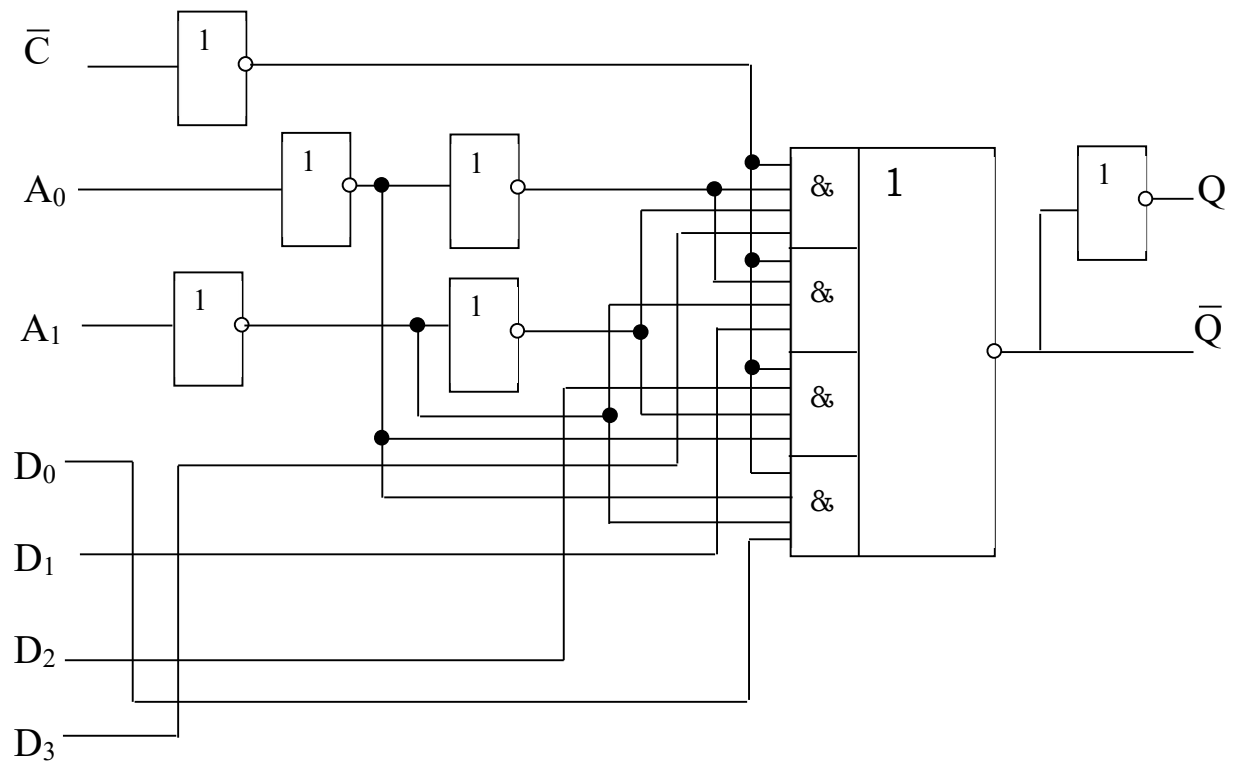
Stroblaşdırıcı siqnal verildikdə ( $C=1$ ) çıxışa i-ünvanı ünvan girişlərindən 2-lik formada verilən  $D_i$  informasiya girişinin məntiqi səviyyəsi ötürülür. Belə ki, məsələn  $A_1A_0$  girişlərinə 11<sub>2</sub> verildikdə Q çıxışına 3 ünvanı- yəni  $D_3$  informasiya girişi ötürülür.

$$A_1A_0=11_2=3_{10}$$

Cədvəldən aşağıdakı məntiqi ifadəni yazmaq olar:

$$Q = \left( D_0 \cdot \overline{A_1} \cdot \overline{A_0} \vee D_1 \cdot \overline{A_1} \cdot A_0 \vee D_2 \cdot A_1 \cdot \overline{A_0} \vee D_3 \cdot A_1 \cdot A_0 \right) \cdot C$$

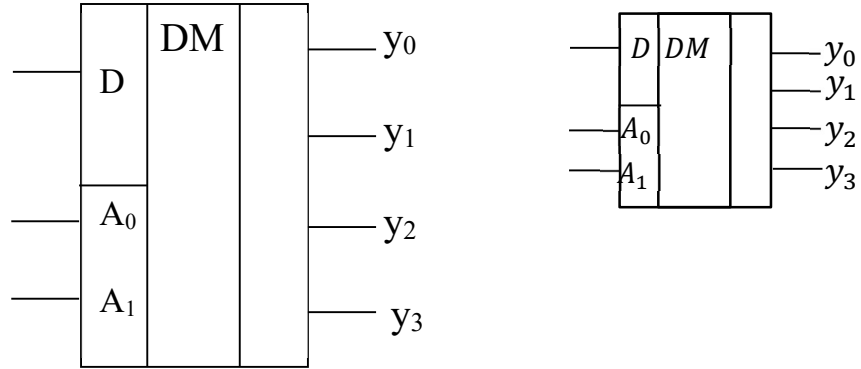
Bu ifadəyə əsasən multipleksorun prinsipial sxemi aşağıdakı kimi olacaqdır.





## Demultipleksor.

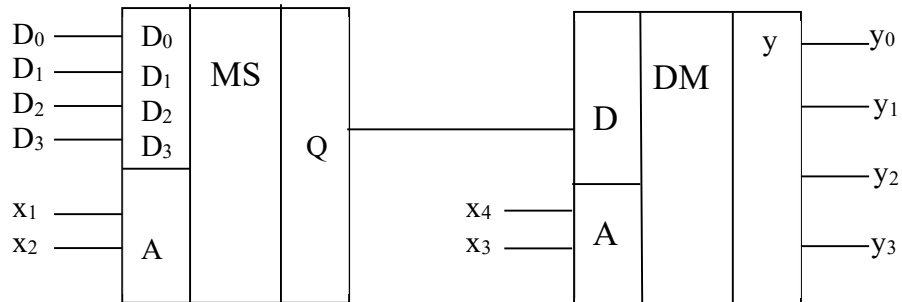
Multipleksorlardan fərqli olaraq DM+ yalnız 1informasiya girişinə və bir neçə çıxışa malik rəqəmli qurğudur. DM-or MS-unəksi olaraq informasiya girişindəki kodu ünvanın kodundan asılı olaraq çıxış kanallarından hər hansı birinə ötürür. Praktiki olaraq rəqəm qurğularda istifadə olunan DM+ 1-4, 1-8, 1-16 olurlar. Funksional sxemlərin təhlilində əsas etibarilə 1-4 prinsipli DM+dan istifadə olun-duğundan onun qurulmasına baxaq. İntegral sxemlərdə 1-4 DM-un mikrosxemi aşağıdakı kimi göstərilir.



DM-un həqiqətler cədvəli aşağıdakı kimi olar.

Ünvan girişləri		Çıxışlar			
A <sub>1</sub>	A <sub>0</sub>	y <sub>0</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

MS-u DM-lə birləşdirərək elə bir qurğu almaq olar ki, verilən ünvana uyğun olan girişlərdən birini çıxışlardan birinə qoşmaq olsun. Bununla da girişlə çıxışın birləşdirilməsinin istənilən kombinasiyasını almaq olar.



Məsələn:

$$x_1 = 1, x_2 = 0$$

$x_3 = 0, x_4 = 0$  olduqda  $D_2$  girişi  $y_0$  çıxışına qoşulmuş olur.

Qeyd edək ki, əyər DM-in girişinə  $D = 1$  konstantası verilsə, onda verilən ünvana uyğun seçilən çıxışda məntiqi 1 olacaq, qalan çıxışlarda isə məntiqi 0 səviyyəsi alınacaq. Bu vaxt yerinə yetirildiyi funksiyaya görə DM-or deşifratora çevrilir.

Doğruluq cədvəlinə əsasən çıxışlara uyğun məntiqi ifadələri yazsaq aşağıdakıları almış olarıq:

$$y_0 = D_0 \bar{A}_0 \bar{A}_1$$

$$y_2 = D_0 \bar{A}_0 A_1$$

$$y_1 = D_0 \bar{A}_0 A_1$$

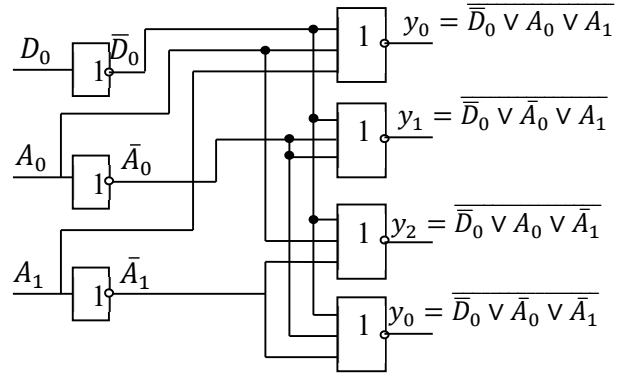
$$y_3 = D_0 A_0 A_1$$

$$y_0 = \overline{\overline{D_0 \bar{A}_0 \bar{A}_1}} = \overline{D_0 \vee A_0 \vee A_1}$$

$$y_1 = \overline{\overline{D_0 \bar{A}_0 A_1}} = \overline{D_0 \vee A_0 \vee A_1}$$

$$y_2 = \overline{\overline{D_0 \bar{A}_0 A_1}} = \overline{D_0 \vee A_0 \vee A_1}$$

$$y_3 = \overline{\overline{D_0 A_0 A_1}} = \overline{D_0 \vee A_0 \vee A_1}$$



Yuxarıdakı ifadəyə əsasən sxemi quraq.

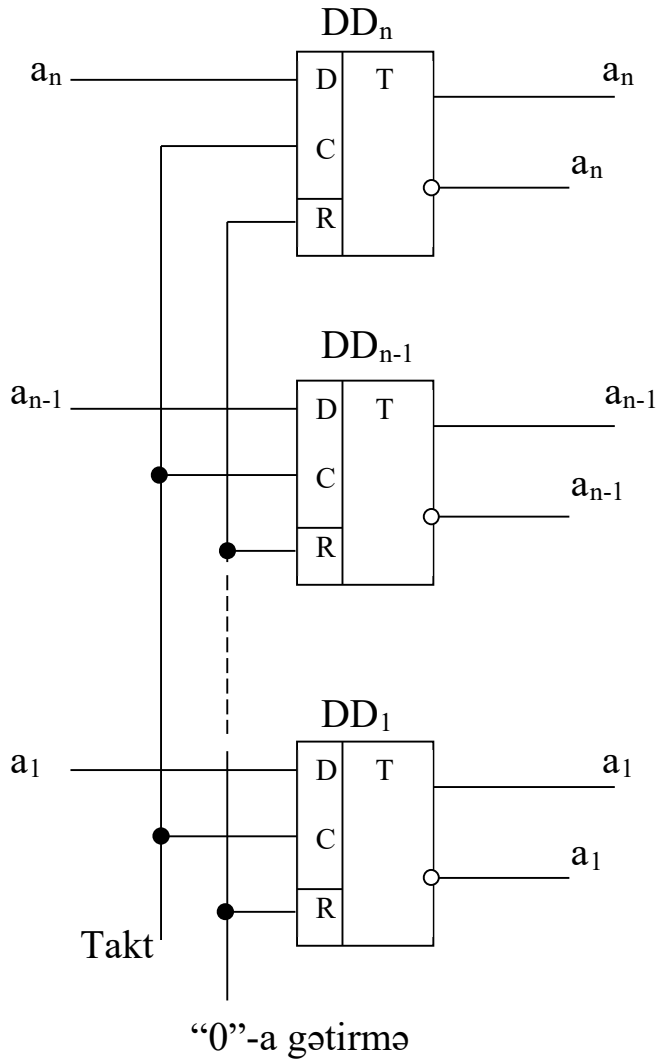
## Reqistrlər.

Reqistrlər rəqəmli qurğu olub 2-lik kod informasiyasının qəbul olunub saxlanılmasında, əks kodun alınmasında, kodların sürüşdürülməsində (sola, sağa), digər qurğuların idarə olunmasında və s-də istifadə olunur. Reqistrlərin yığılmasında triggerlərdən istifadə olunur.  $n$  dərəcəli sözün saxlanılmasında  $n$  sayda trigger götürülür. İşləmə xüsusiyyətinə görə reqistrlər 2 növə ayrılır.

- 1) Paralel təsirli reqistrlər.
- 2) Ardıcıl təsirli reqistrlər.

Ardıcıl təsirli reqistrlərdə kod sözünün dərəcələri ardıcıl olaraq qəbul olunub oxunulur.

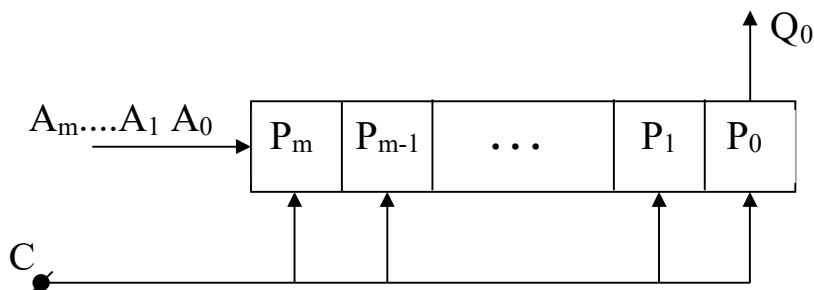
Paralel təsirli reqistrlərdə isə kod sözünün bütün dərəcələri eyni zamanda qəbul olunub, oxunulur. Belə reqistr sxeminin yığılmasında D tipli triggerlərdən istifadə olunur. Giriş kodu uyğun olaraq  $DD_1$ - $DD_n$  triggerlərinin D girişinə verilir. C girişinə takt idarəetmə impulsu təsir etdikdə triggerlər eyni zamanda giriş koduna uyğun vəziyyətə gətirilir. Giriş kodun əks qiymətini oxumaq istədikdə triggerlərin inkar çıxışından istifadə olunur.



## Reqistrlər

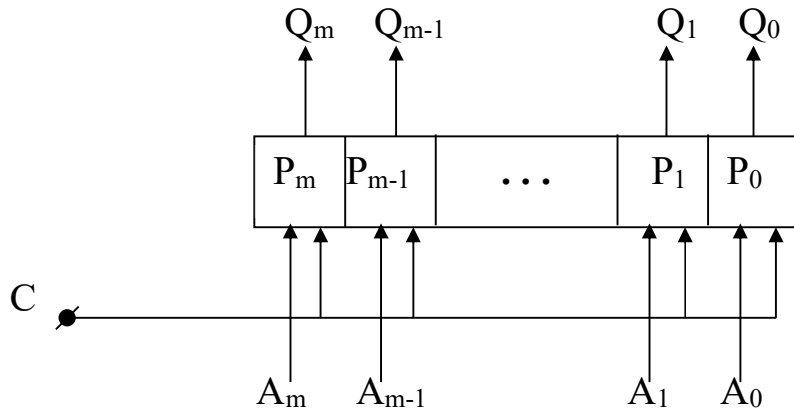
Kompyuterlərdə reqistrlər demək olar ki, maşını təşkil edən bütün qurğularda istifadə olunur. Reqistrlər sözlərin yadda saxlanması, qəbulu və lazım gəldikdə hər hansı bir obyektə digər obyektə ötürmək üçün istifadə olunur. Bununla yanaşı reqistr sxemləri 2-lik kod çevriciləri kimi də istifadə oluna bilər. Reqistrdə yerinə yetirilən əməliyyat yalnız 2-lik modluna görə aparılır. Reqistr sxemləri bizə məlum olan triggerlər üzərində qurulur və reqistrlərin mərtəbələrinin sayı həmin reqistrləri təşkil edən triggerlərin sayı ilə müəyyən olunur. Yəni trigger əgər 1 bit informasiyanı yadda saxlayırsa, N mərtəbəli reqistrdə m bit informasiya yazmaq mümkün olur. İşləmə prinsipinə görə reqistrlər **3** növ olur:

**1. Ardıcıl prinsipli reqistrlər.** Praktikada belə prinsiplə işləyən reqistr sxemlərinə sürüşdürücü reqistr sxemləri də deyilir. Ona görə ki, bu halda reqistrə yazılacaq ədədin kodu reqistrə ardıcıl olaraq kodun ən kiçik mərtəbəsindən başlayaraq yazılır və hər bir mərtəbəyə uyğun bir takt signalı verilir. Ümumi şəkildə ardıcıl prinsipli reqistr sxemini aşağıdakı kimi təsvir etmək olar.



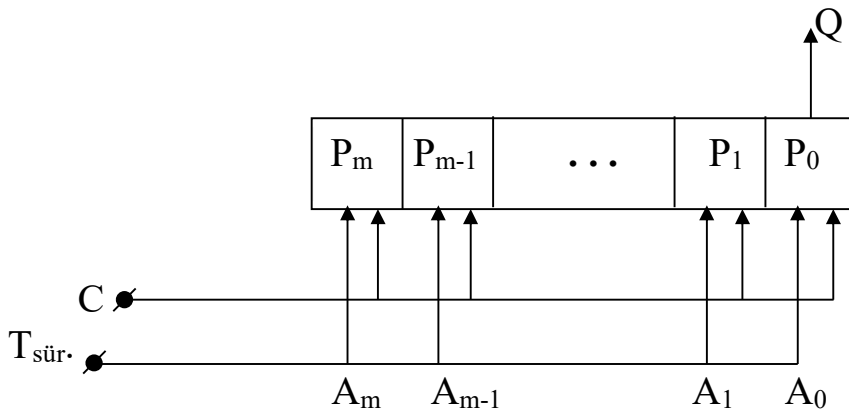
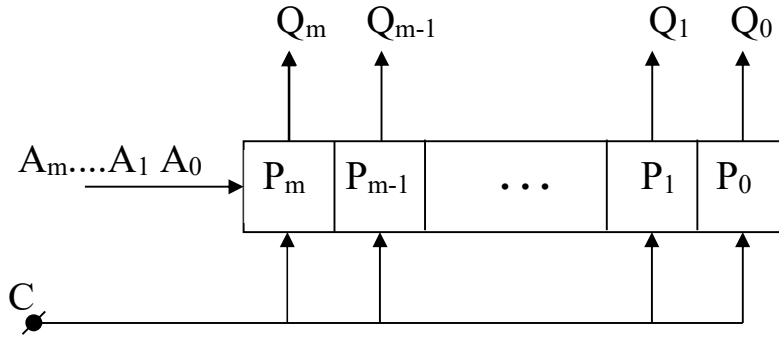
Bu üsulla informasiyanı (ədəd kodunu) reqistrə yazdıqda reqistrin sonuncu 1-ci mərtəbədə başlayaraq əvvəlcə kodun ən kiçik mərtəbəsi ( $A_0$ ) ardıcıl olaraq  $C$  takt signalı verməklə reqistrin  $P_m$ -ci mərtəbəsində yazılır. Növbəti sinxroimpuls və ona uyğun ədədin 2-ci mərtəbəsi  $A_1$  reqistrə verildikdə  $P_m$ -də yazılmış olan ədədin birinci mərtəbəsi, yəni  $A_0$   $P_{m-1}$ -ə sürüşdürülür və  $P_m$  mərtəbəsində  $A_1$  yazılır. Beləliklə,  $m$  mərtəbəli reqistrə  $m$  mərtəbəli ədədin yazılması üçün  $m$  ədəd sinxrotakt impulsu vermək lazımdır.  $m$ -ci takt impulsu verildikdə ədədin ən böyük mərtəbəsi ardıcıl olaraq  $P_m$  mərtəbədə yazılır. Onun ən kiçik mərtəbəsi isə ( $A_0$ )  $Q_0$  çıxışından ardıcıl oxunur.  $m$ - mərtəbəli ədədin oxunması üçün ardıcıl olaraq hər ədəd mərtəbəsinə uyğun  $m$  ədəd sinxroimpuls verilir və hər sinxroimpulsdan sonra  $Q_0$  çıxışından növbəti sinxroimpulsa uyğun ədəd mərtəbəsi oxunur. Nəhayət,  $m$  ədəd sinxroimpulsdan sonra reqistrdə yazılmış bütün mərtəbələr ardıcıl oxunur və reqistr təmizlənmiş olur.

**2-ci üsul paralel prinsipli reqistrlərdir.** Bu reqistr sxemlərində yazılacaq ədədin kodu paralel olaraq reqistrin uyğun girişlərinə verilir və bu halda yalnız bir ədəd takt sinxrosignalı vermək tələb olunur.



Bu halda reqistrə paralel yazılan ədədin kodu reqistrdən paralel olaraq oxu-nur. Belə reqistrlərə yaddaşlı reqistrlər də deyilir.

**3-cü qarışıq prinsipli reqistrlərdir.** Bu prinsipli reqistrlərdə informasiya ya reqistrə ardıcıl yazılır və paralel oxunur, ya da əksinə paralel yazılır, ardıcıl oxu-nar.



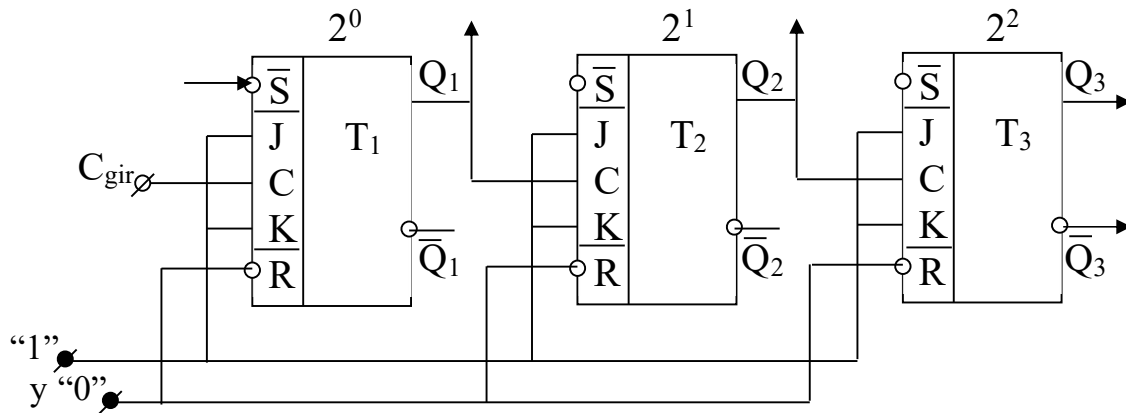
Göründüyü kimi belə prinsipli reqistrlər sxemində yazılacaq ədəd ancaq bir istiqamətdə sürüşdürülmüş olur. Lakin əlavə sürüşdürücü, idarəedici siqnalların köməyiylə reqistrdə informasiyanı həm sağa, həm də sola sürüşdürmək mümkün olur. İnformasiyanı həm sağa, həm də sola sürüşdürmək üçün praktiki olaraq reversiv reqistrlərdən istifadə olunur. Göstərilən prinsipli reqistrlər əsasən RS, D, JK triggerləri üzərində qurulur.

## Sayğaclar. Cəmləyici ikilik sayğac

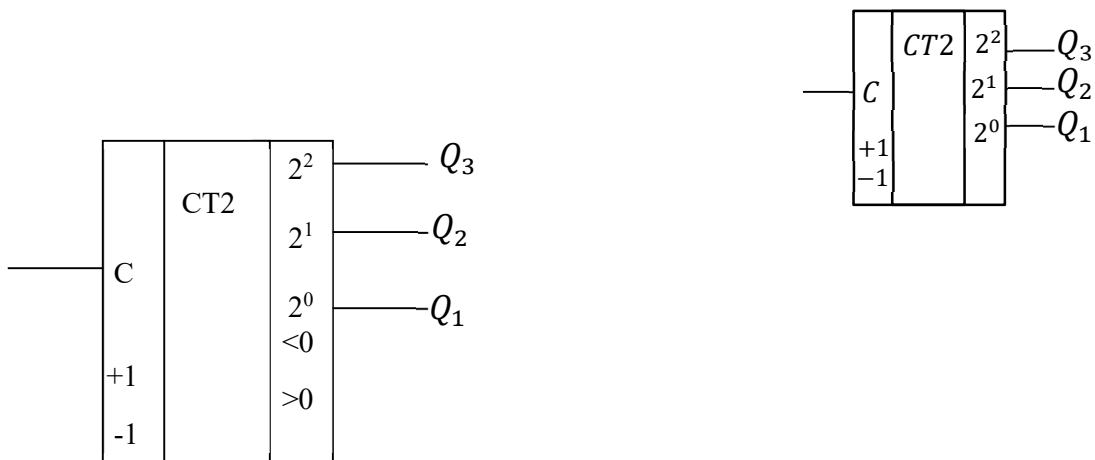
EHM+də sayğaclardan əsas etibarilə idarəetmə qurğularında istifadə olunur. Belə ki, sayğacla impulsların sayılması, əməllərin ünvanlarının təşkili, əməliyyatların y/yetirilməsi ardıcılığının sayılması üçün istifadə olunur. Sayğac sxemləri də reqistr sxemlərində olduğu kimi triggerlər üzərində qurulur və sayğacın mərtəbələrinin sayı onda istifadə olunan triggerlərin sayı ilə müəyyən olunur. İşləmə prinsipinə görə  $S+2$  yerə bölünür: Sadə və reversiv sayğacla.

Sadə sayğacla öz növbəsində cəmləyici və çıxıcı olurla. Reversiv sayğacla həm toplama, həm də çıxma əməliyyatını y/yetirir. Sayma modluna görə sayğacla 2-lik, 10-luq, 2-10-luq tipli olurla. Sayğac sxemləri əsas etibarilə RS, JK, D sinxron triggerlər üzərində qurulur.

Sayğac sxemlərində triggerlər say rejimində işləyirlə. Ona görə də ardıcıl keçidli cəmləyici sayğacı qurmaq üçün JK triggerinin J və K girişlərini birləşdirib ona "1" siqnalı verək. "1" -ə uyğun Q çıxışını isə növbəti kiçik mərtəbənin triggerinin sinxrogirişi ilə birləşdirmək lazımdır. Bu halda sxem aşağıdakı kimi olacaq.



Funksional olaraq aşağıdakı kimidir.



Sxemdə istifadə olunan y “0” idarəedici siqnalı sayğacın məzmununu “0” vəziyyətinə gətirmək üçün istifadə olunur. Lakin bu o demək deyil ki, sayğaca giriş siqnalı yazmazdan qabaq hökmən ilk vəziyyətə gətirilməlidir. Ola bilər ki, sayğac mərtəbələrində yazılmış hər hansı 1 ədəd kodunun üzərinə növbəti im-pulslar sayılması tələb olunsun. Sayılan impulslar sayğacın C girişindən ardıcıl olaraq verilir.  $\bar{S}$  girişi vasitəsilə impulsların paralel olaraq uyğun mərtəbələrdə yazılması və sayğacın hər hansı bir mərtəbəsində informasiyanın yenidən yazılması üçün istifadə olunur. 1-ci sinxroimpuls verildikdə  $T_1$  triggeri vahid vəziyyətə keçir. Bu halda  $T_2$  və  $T_3$ -ün məzmunu dəyişməz qalır. Növbəti sinxroimpulsda  $T_1$ -də yazılmış “1”  $Q_1$  çıxışı ilə  $T_2$ -ni “1” vəziyyətinə gətirir.  $T_1$ -də “0” yazılır.  $T_3$  yenə öz vəziyyətini dəyişmir. C girişinə 3-cü sinxroimpuls verildikdə  $T_1$ -ə “1”,  $T_2$ ,  $T_3$  öz vəziyyətini saxlayır. Yəni sayğacda 011 yazılmış olur. 4-cü sinxroimpuls verildikdə  $T_3$ -ə “1” yazılır,  $T_1$ ,  $T_2$ -də “0” olur. Nəhayət 7-ci sinxroimpuls verildikdən sonra sayğacın bütün mərtəbələrində “1” yazılmış olur. Yəni 111 olur. Əgər 8-ci sinxroimpuls verilərsə, onda bütün mərtəbələrdə “0” yazılmış olmalıdır. Göründüyü kimi ardıcıl keçidli cəmləyici sayğac sxemlərində mərtəbələrin sayını artırmaq üçün triggerlərin sayını artırılması lazım gəlir. Lakin bu bir tərəfdən sayma modlunun artmasına səbəb olursa, digər tərəfdən sayğacın işləmə sürətini aşağı düşməsinə səbəb olur. Ona görə də 1-ci mərtəbənin triggeri növbəti mərtəbənin triggerinə nəzərən daha böyük sürətlə işləmiş olur ki, bu da sxemin girişində böyük güc sərfinə gətirib çıxarır.

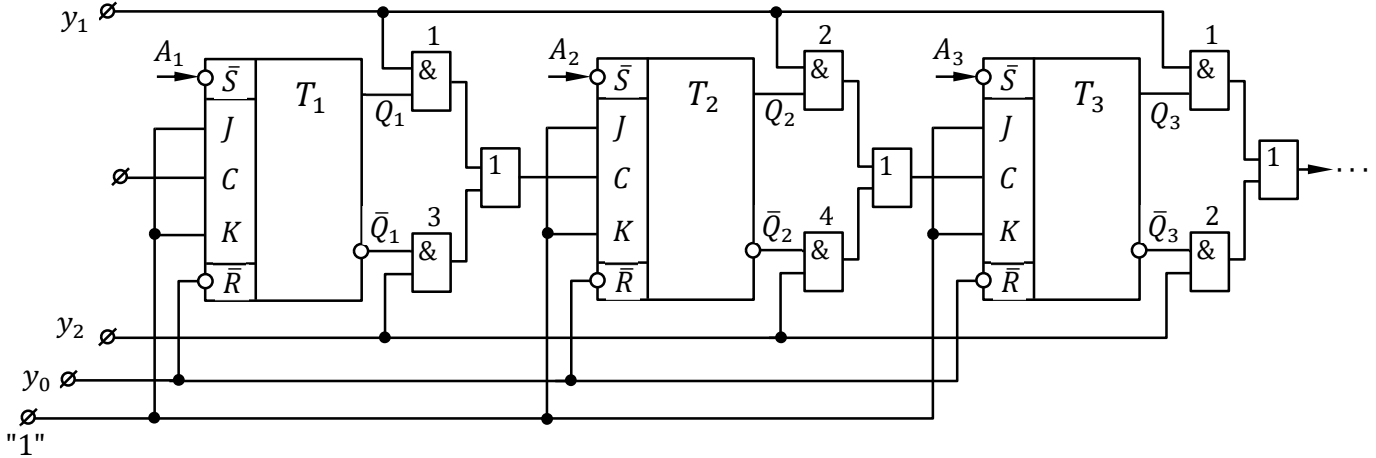
Sayğac sxemlərində ümumi gecikmə vaxtı aşağıdakı kimi təyin edilə bilər.

$$T_{\text{gec.}} = T_{\text{trg.}} \cdot N$$

Burada  $T_{\text{tr.gec.}}$  – 1 triggerdə gecikmə vaxtı, N- mərtəbələrin ümumi sayıdır. Ardıcıl keçidli cəmləyici sayğac sxeminin sürətini artırmaq üçün bəzi hallarda müxtəlif parametrlə triggerlərdən istifadə olunur. Lakin bu halda idarəedici qurğuda müxtəlif mürəkkəb məntiqi dövrlərin yaranmasına səbəb olur. Bu səbəbdən paralel prinsipli sayğac sxemləri istifadə olunur.

## Reversiv sayğaclar

Praktiki olaraq sadə funksional sxemlərin tərtib olunması zamanı sayğac sxemlərindən istifadə olunması lazım gəldikdə əsas etibarlı ilə reversiv sayğaclardan istifadə olunması məqsədəuyğun sayılır. Reversiv sayğac sxemləri üzərində impulsların həm cəmlənməsini, həm də çıxılmasını yerinə yetirmək mümkün olur. JK triggeri üzərində qurulmuş reversiv sayğac sxemi aşağıdakı kimidir.



Sxemdə göstərilən  $y_1$  və  $y_2$  idarəedicilərdir. Belə ki,  $y_1$  reversiv sayğacı cəmləmə rejimində işlətmək üçün,  $y_2$  isə çıxma rejimində işlətmək üçün istifadə olunur. Hər iki halda sayılan impuls  $C$  girişinə verilməlidir. Reversiv sayğacın cəmləmə rejimində işləməsinə baxaq.

Əvvəlcə  $y_0$  idarə signalı verməklə sayğacı 0 vəziyyətinə gətiririk. Bundan sonra sayılan 1-ci impuls  $T_1$ -in  $C$  girişinə verilir. İmpulsun təsir müddəti qurtardıqdan sonra  $T_1$  1 vəziyyətinə keçir. Digər tərəfdən qeyd etmək lazımdır ki, keçid mərtəbələrinin vəziyyətləri idarə signallarından asılı olduğundan göstərilən hal üçün  $y_1=1$ ,  $y_2=0$  signallarını vermək lazımdır. Bu halda 1-ci və 2-ci VƏ elementləri açıq, 3-cü və 4-cü VƏ məntiq elementləri bağlı vəziyyətdə olmalıdır. Sayılan növbəti 2-ci impuls ardıcıl olaraq  $C$  girişinə verildikdən 1-ci VƏ, VƏ YA dövrəsi ilə  $T_2$ -yə vahid yazılacaq. Bu halda  $T_1$ -ə 0 yazılmış olar. 3-cü sinxroimpuls təsir etdikdən sonra  $T_1$ -ə 1 yazılır,  $T_2$  və  $T_3$  vəziyyətini dəyişmir. Yəni sayğacın göstərişi bu halda 011 olmalıdır. Növbəti 4-cü impulsun təsirindən sonra 1-ci VƏ, VƏ YA dövrəsi ilə  $T_2$ -yə 0 yazılır.  $Q_2$  çıxışından alınan vahid keçid mərtəbəsi 2-ci VƏ, VƏ YA dövrəsi ilə  $T_3$ -ün say girişinə verilir və  $T_3$  1 vəziyyətinə gətirilir. Bu halda  $T_1$  də 0 vəziyyətində olur. Beləliklə 7 impuls təsirindən sonra sayğacın məzmunu 111 olmalıdır. Əgər bu halda 8-ci impuls təsir edərsə onda sayğacın göstərişi 0 olmalıdır.

İndi isə reversiv sayğacın çıxma rejimində işləməsinə baxaq. Yenə reversiv sayğacın çıxma rejimində işləməsi üçün əvvəlcə  $y_0$  signalı verməklə sayğacı 0 vəziyyətinə gətiririk. Bundan sonra idarə girişinə  $y_1=0$ ,  $y_2=1$  signalı veririk. Bu halda 1-ci və 2-ci VƏ məntiq elementləri bağlı, 3-cü və 4-cü VƏ məntiq elementləri açıq olur. Daha sonra çıxılması tələb olunan hər hansı bir ədədin kodu (məs. 111) paralel olaraq  $A_1$ ,  $A_2$ ,  $A_3$  girişlərindən sayğaca yazılır. Çıxılacaq impuls ardıcıl olaraq bu halda da  $C$  girişinə verilir. 1-ci impulsun təsirindən sonra  $T_1$  0 vəziyyətinə gətiriləcək, qalan  $T_2$  və  $T_3$  triggerləri isə əvvəlki vəziyyətlərində qalacaq. 2-ci impulsun təsirindən sonra  $Q_1$  çıxışı 3-cü VƏ məntiq elementi açıq



olduğundan  $T_2$ -yə 0 yazılmış olur. Bu halda  $T_1$ -ə 1 yazılır. Yəni sayğacın göstərişi 100 olmalıdır. Beləliklə 7 ardıcıl impulsun təsirindən sonra sayğacın göstərişi 000 olacaqdır.

## Cəmləyicilər

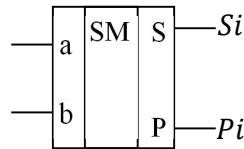
Cəmləyicilər (SM) ədəd kod mərtəbələri üzərində hesabi əməliyyatların y/y-si üçün istifadə olunan kombinasiyon sxemə deyildir. Əgər giriş operandları (verilənləri) və alınan nəticə 2-lik s/s-nə uyğun ədədlər kimi təsvir olunursa, onda belə cəmləyicilər 2-lik cəmləyicilər adlanır. S/s-dən asılı olaraq 2 ədədin cəmlənməsi zamanı 1 takt ərzində aşağıdakı rəqəmlərin toplanması həyata keçirilir:

- 1.I cəmlənən mərtəbənin rəqəmi
- 1.II cəmlənən mərtəbənin rəqəmi
- 3.Qonşu mərtəbəyə keçid rəqəmi.

Girişlərin sayına görə SM+ aşağıdakı növlərə bölünür:

- 1.Yarım cəmləyici
- 2.1 mərtəbəli cəmləyici
- 3.Çoxmərtəbəli cəmləyicilər.

**Yarım cəmləyicilərin** əsas etibarı ilə 2 giriş və uyğun olaraq 2 çıxışı olur. Girişlərə toplanan I və II ədədlər verilir və bu halda kiçik mərtəbədən keçid rəqəmi nəzərə alınmır. Yarım SM-in sxemi aşağıdakı kimi göstərilir.

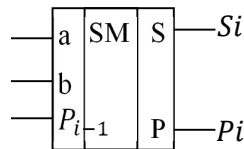


Cəmləyicinin çıxışında alınan çəm mərtəbəsi  $a_i$  və  $b_i$  toplananlarına uyğun olaraq:

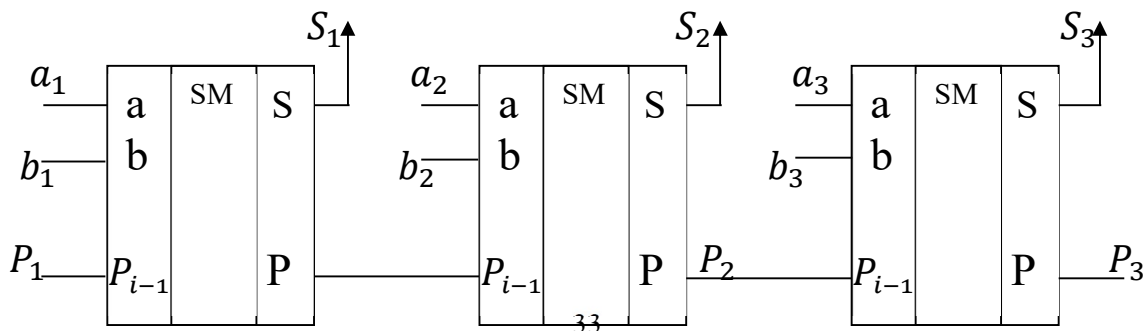
$$S_i = a_i \oplus b_i$$

Qonşu mərtəbəyə keçid  $P_i = a_i b_i$  kimi təyin edilir.

**1 mərtəbəli cəmləyicilər** yarım SM+dən fərqli olaraq 3 ədəd giriş və 2 ədəd çıxışı olur.



Sxemdən görünür ki, 1 mərtəbəli SM+də giriş verilənlərindən başqa həm də kiçik mərtəbədən keçid rəqəmi də nəzərə alınır. **Çox mərtəbəli cəmləyicilər** əsas etibarı ilə 1 mərtəbəli SM+ üzərində qurulur və ilişmə prinsipinə görə ardıcıl və paralel prinsipli olurlar. 1 mərtəbəli SM+ üzərində qurulmuş çox mərtəbəli SM sxemi aşağıdakı kimidir.



Sxemdən göründüyü kimi SM 3 mərtəbəli SM-dir.

Ardıcıl üsulla cəmləmə əməliyyatını y\y-ək üçün cəmlənməsi tələb olunan verilənlər uyğun şəkildə qruplaşdırılır və hər bir takt ərzində toplanan mərtəbələr-dən yalnız və yalnız uyğun qruplar cəmləyicinin uyğun girişinə verilir.

Ardıcıl girişli çox mərtəbəli SM+ və paralel girişli SM+ 1 mərtəbəli SM+ üzərində qurulduğundan göstərilən SM sxem üzərində həm də paralel prinsiplə toplama əməliyyatını y\y-ək olar. Bu halda kod sözünün bütün mərtəbələri paralel olaraq SM-n uyğun girişinə verilir. Cəmləmə əməliyyatı 1 takt ərzində y\y-lir. Pa-ralel SM+də keçid mərtəbəsi növbəti qonşu mərtəbəyə ya paralel, ya da ardıcıl olaraq ötürülə bilər. SM+ əsas etibarilə qurulma prinsipinə görə **yaddaşlı və yaddaşsız SM+ə bölünür.**

**Yaddaşsız SM+ə** bəzən kombinasiyon tipli SM+ deyilir. Belə SM+ əsas etibarilə potensial məntiq elementləri üzərində qurulur və heç bir yaddaşa məlik olmur. Yaddaşsız SM+ aşağıdakı mikroəməliyyatı y\y-rir.

$$S_i = A + B$$

Yaddaşsız SM+dən fərqli olaraq **yaddaşlı SM+də** yaddaş elementi kimi triggerlərdən istifadə olunur. Belə SM+də cəmləmə əməliyyatı 2-lik moduluna görə y\y-lir. Yaddaşlı SM+ aşağıdakı mikroəməliyyatı y\y-rir.

$$S_i = S + A$$

İşləmə prinsipinə görə SM+ **asinxron və sinxron** girişli olur.

**Sinxron tipli SM+də** əvvəlcədən cəmləmə taktı əvvəlcədən müəyyənləşmiş olur. Bu takt (vaxt) sabitdir və toplananların mərtəbəsindən asılı deyildir.

**Asinxron tipli SM+də** cəmləmə əməliyyatı qurtardıqdan sonra əməliyyatın sona çatması haqqında əlamət signalı hasil olunur. Cəmləmə əməliyyatında istifadə olunan kodlardan asılı olaraq SM+ 2-lik və 2-10-lik SM+i olurlar.

Ardıcıl keçidli 1 mərtəbəli kombinasiyon cəmləyicinin işinə baxaq.

2 hesabi verilənlər üzərində əməliyyat aparılmasına müvafiq olaraq n mərtəbəli:

$$A = a_1, a_2, \dots, a_n$$

$$B = b_1, b_1, \dots, b_n$$

Ədədlərin toplanması aşağıdakı mərtəbə cəmlərinin hesablanmasına gətirilir:

$$q_i S_i = a_i + b_i + P_i$$

$q_i S_i$  - 2 mərtəbəli 2-lik ədədi,  $q_i$  - i-ci mərtəbədə keçid rəqəmi,  $S_i$  -  $a_i$  və  $b_i$  mərtəbələrinə uyğun alınan cəm,  $P_i$  -qonşu mərtəbəyə alınan keçiddir.

1 mərtəbəli cəmləyicinin iş prinsipini izah etmək üçün aşağıdakı doğruluq cədvəlinə baxaq.

$P_i$	$a_i$	$b_i$	$S_i$	$P_{i+1} (q_i)$
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

Doğruluq cədvəlinə əsasən  $i$ -ci mərtəbədə uyğun cəm və növbəti böyük mərtəbəyə keçid üçün aşağıdakı məntiqi ifadəni yazmaq olar:

$$P_{i+1} = \bar{P}_i a_i b_i \vee P_i \bar{a}_i b_i \vee P_i a_i \bar{b}_i \vee P_i a_i b_i$$

$$S_i = \bar{P}_i \bar{a}_i b_i \vee \bar{P}_i a_i \bar{b}_i \vee P_i \bar{a}_i \bar{b}_i \vee P_i a_i b_i$$

$$P_{i+1} = (\bar{P}_i a_i b_i \vee P_i a_i b_i) \vee (P_i \bar{a}_i b_i \vee P_i a_i b_i) \vee (P_i a_i \bar{b}_i \vee P_i a_i b_i) =$$

$$= a_i b_i (\bar{P}_i \vee P_i) \vee P_i b_i (\bar{a}_i \vee a_i) \vee P_i a_i (\bar{b}_i \vee b_i) = a_i b_i \vee P_i b_i \vee P_i a_i$$

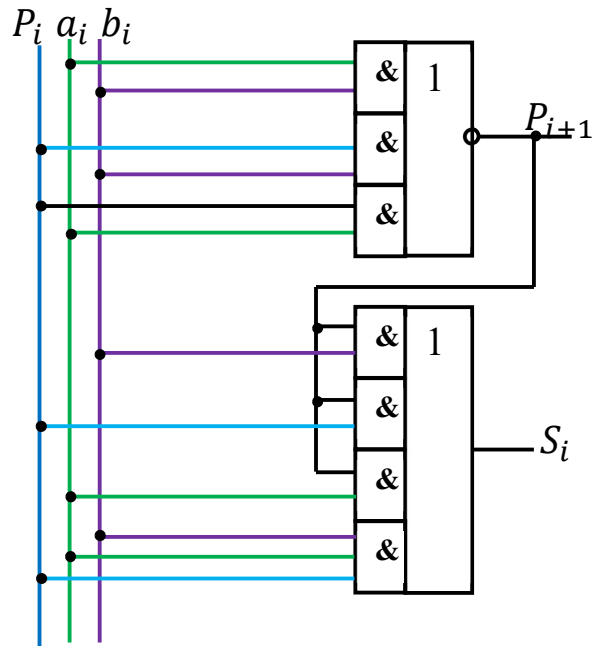
$$S_i = (\bar{P}_i \bar{a}_i b_i \vee b_i \bar{b}_i \bar{P}_i \vee b_i \bar{b}_i \bar{a}_i) \vee (\bar{P}_i a_i \bar{b}_i \vee a_i \bar{a}_i \bar{P}_i \vee a_i \bar{a}_i \bar{b}_i) \vee (P_i \bar{a}_i \bar{b}_i \vee \bar{P}_i \bar{a}_i P_i \vee P_i \bar{P}_i \bar{b}_i)$$

$$\vee b_i a_i P_i$$

$$= b_i (\bar{P}_i \bar{a}_i \vee \bar{b}_i \bar{P}_i \vee \bar{b}_i \bar{a}_i) \vee a_i (\bar{P}_i \bar{b}_i \vee \bar{a}_i \bar{P}_i \vee \bar{a}_i \bar{b}_i) \vee P_i (\bar{a}_i \bar{b}_i \vee \bar{P}_i \bar{a}_i \vee \bar{P}_i \bar{b}_i)$$

$$\vee P_i a_i b_i = P_{i+1} (b_i \vee P_i \vee a_i) \vee P_i a_i b_i$$

$P_{i+1}$  və  $S_i$  üçün aldığımız minimallaşdırılmış dizyuktiv normal forma ifadəsinə görə kombinasiyon cəmləyicinin sxemi aşağıdakı kimi olar.



## Kod komparatorları

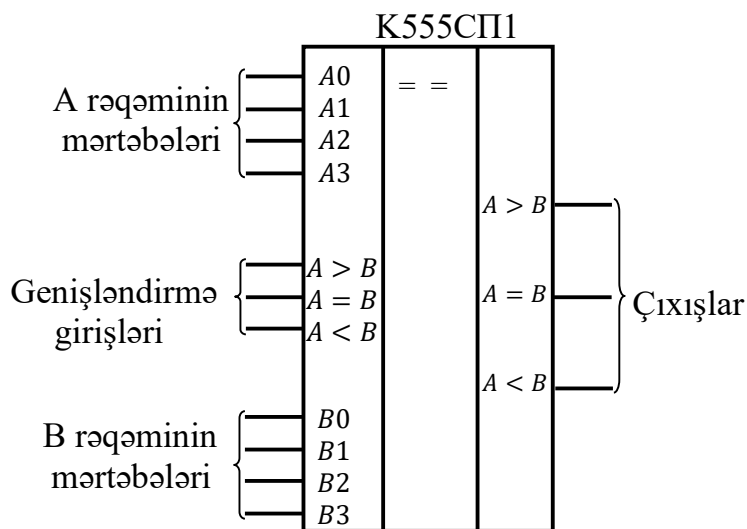
Rəqəmli komparatorlar – kombinasiyalı qurğu olub ikilik sözləri müqayisə etmək üçündür. Komparator iki ikilik sözlər üzərində aşağıdakı əməliyyatları y/y:

$F = (A = B)$  - A və B sözlərinin bərabərliyi;

$F = (A > B)$  - A sözü B sözündən böyükdür;

$F = (A < B)$  - A sözü B sözündən kiçikdir.

Komparatora misal olaraq iki dördmərtəbəli sözün müqayisə sxemi olan K555CП1 mikroseminə baxaq. Əsas məsələ üç mümkün olan vəziyyətdən birini:  $A < B, A = B, A > B$  təyin etməkdir. Bərabərliyin əsas göstəricisi iki ikilik rəqəmin bütün mərtəbələrinin uyğun olmasıdır. Əgər iki rəqəm eynidirsə müqayisə sxeminin çıxışı “1” məntiqi səviyyəyə gətirilir, əks halda çıxış məntiqi “0” səviyyəsində olur.



Şəkil1.

K554CП1 mikrosxemasının A və B rəqəmlərinin dörd ( $A0, B0, A1, B1, A2, B2, A3, B3$ ) müqayisə girişləri və üç əlavə keçid girişləri  $A < B, A = B, A > B$  vardır.

Bərabərliyə görə müqayisə qurğusu hər iki sözün eyniadlı mərtəbələri üzərində qurulur. Söz ovaxt bərabər olur ki, eyniadlı mərtəbələr bərabər olsun, yəni hər ikisində 0 və 1-lər eyni olsun.

Bir ikilik mərtəbənin müqayisəsinə baxaq.

Cədvəl1.1

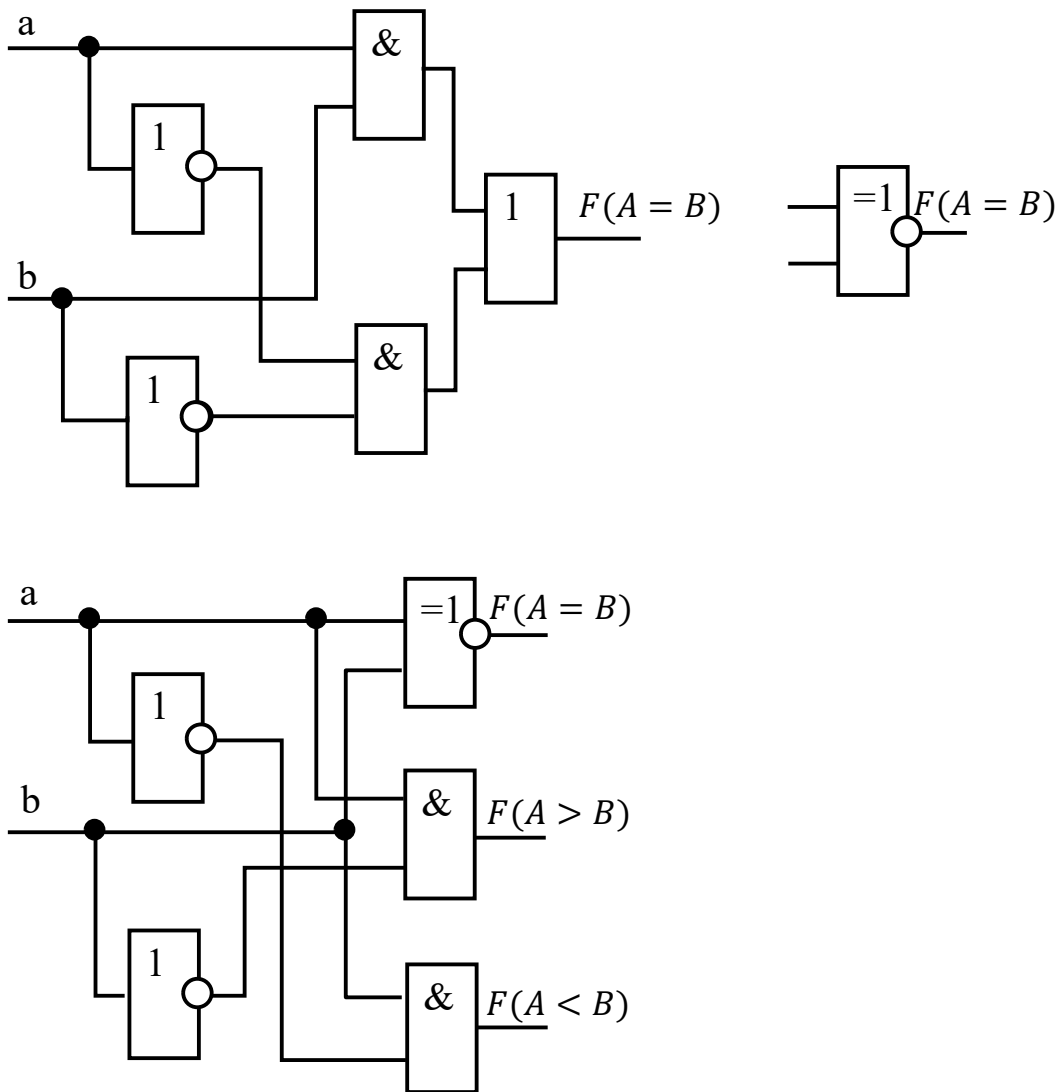
Birmərtəbəli söz komparatorunun həqiqətlər cədvəli

A	B	$F = (A = B)$	$F = (A > B)$	$F = (A < B)$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Cədvəl 1.1-dən görünür ki, giriş siqnalının istənilən kombinasiyasında komparatorun çıxışında ancaq bir aktiv məntiqi signal formalaşır. Cədvəl 1.1-dən aşağıdakı məntiqi ifadəni almaq olar:

$$\begin{aligned}
 F &= (A = B) = ab \vee \bar{a}\bar{b}; \\
 F &= (A > B) = a\bar{b}; \\
 F &= (A < B) = \bar{a}b.
 \end{aligned}
 \tag{1}$$

(1) ifadəsini VƏ, VƏ YA, YOX məntiq elementləri üzərində qurmaq olar.  $F(A = B)$  funksiyasına daha geniş baxaq. Bu funksiya sərbəst mənə daşıyır və rəqəmli qurğularda geniş tətbiq olunur, “VƏ YA- YOX –u inkar edir”, 2-lik moda görə cəm adlanır (şəki 2). (1) ifadəsinə uyğun olaraq bir mərtəbəli komparator sxemi aşağıdakı kimidir.



Şəkil 2

Qeyd edək ki,  $F(A = B)$ ,  $F(A > B)$ ,  $F(A < B)$  üç funksiyasından ikisi heç vaxt eyni vaxta vahid mənasını ala bilməz. Ona görə də onlardan ancaq ikisini tətbiq edib, üçüncüsünü isə məlum olan ikisindən almaq lazımdır. Onlar öz aralarında aşağıdakı qanunauyğunluqla əlaqəlidir:

$$\begin{aligned} F(A = B) &= \bar{F}(A > B) \cdot \bar{F}(A < B); \\ F(A > B) &= \bar{F}(A = B) \cdot \bar{F}(A < B); \\ F(A < B) &= \bar{F}(A = B) \cdot \bar{F}(A > B). \end{aligned} \quad (2)$$

İki sözün bərabərliyini yoxlamaq hər bir mərtəbənin bərabər olması deməkdir.  $F(A > B)$  “böyükdür” –ü yoxladıqda böyük mərtəbələr üstünlük təşkil edir. Kiçik mərtəbələr, böyü mərtəbələr bərabər olduqda yoxlamaq lazımdır. Sözün  $i$ -ci mərtəbəsinin bərabərliyini  $r_1 = F(a_i = b_i)$  kimi qeyd edək. Onda ikimərtəbəli sözün  $A > B$ -yə yoxlanılması  $a_1 b_1$  böyük mərtəbələrinin yoxlanılmasına gətirib çıxarır.  $a_0 b_0$  kiçik mərtəbələr, böyük mərtəbələr bərabər olduqda yoxlanılır:

$$a_1 = b_1 \rightarrow r_1 = F(a_1 = b_1) = 1.$$

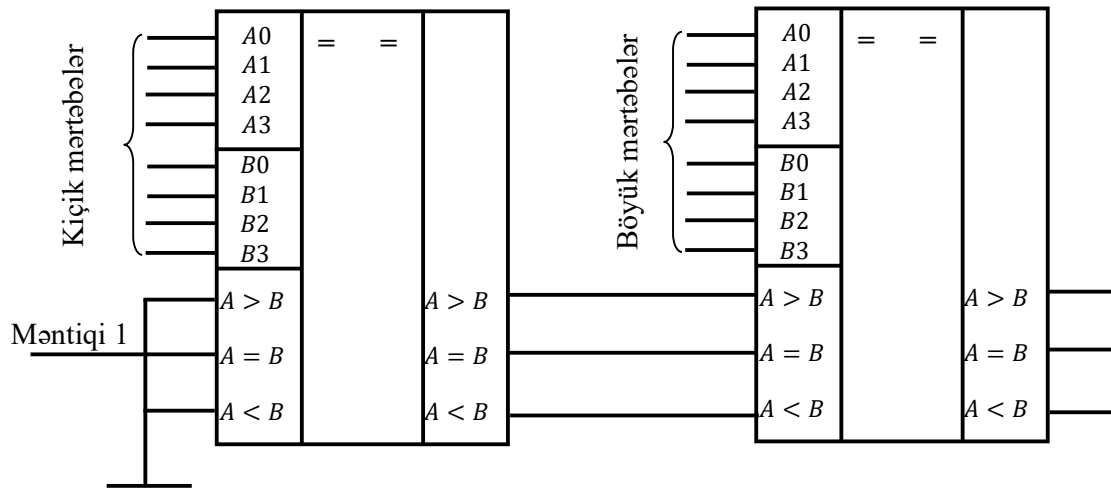
Ona görə də ikimərtəbəli  $A(a_0, a_1), B(b_0, b_1)$  söz üçün  $A > B$  yoxlama formulasını yazaq:

$$F(A > B) = a_1 \bar{b}_1 \vee r_1 a_0 \bar{b}_0. \quad (3)$$

İstənilən  $n$  mərtəbəli söz üçün (3) tətbiq etsək, onda alarıq:

$$F(A > B) = a_{n-1} \bar{b}_{n-1} \vee r_{n-1} a_{n-2} \bar{b}_{n-2} \vee \dots \vee r_{n-1} r_{n-2} r_{n-3} \dots r_1 a_0 \bar{b}_0 \quad (4)$$

Qeyd etdiyimiz kimi, K554CП1 mikrosxemi dördmərtəbəli sözü müqayisə etmək üçündür. Daha böyük mərtəbəli müqayisə etmək üçün bir neçə integral sxemli komparatorları birləşdirərək qururlar (şəkil 3).



Şəkil 3.

Kodlardan birinin dəyişdirilməsi komparatorların çıxışında qeyri – müəyyən vəziyyət yaradır. Buna səbəb giriş signalının müxtəlif mərtəbələrdə eyni zamanda dəyişməsinin baş verməsidir. Bu zaman çıxışda qısa xəta impulsları əmələ gəlir. Xətalərin olmaması üçün sinxronizasiyadan və stroblaşdırmadan istifadə olunur.

Onu da nəzərə almaq lazımdır ki,  $n$  mikrosxemin birləşməsindən (şəkil 2) istifadə etdikdə signalın ümumi gecikməsi  $n$  dəfə artır. Kod komparatoru həddən artıq asta işləyən qurğudur.

## Mikroprosessorun tipik strukturu

Mikroprosessorun tipik strukturu üç əsas blokdən ibarətdir: hesab- məntiq qurğusu (HMQ), daxili registrlər bloku və idarəetmə qurğusu (şəkil 1). Bu bloklar arasında məlumatın ötürülməsi daxili verilənlər şini vasitəsi ilə həyata keçirilir. HMQ-nun əsas vəzifələrindən biri- məlumatın emalının yerinə yetirilməsidir. Onun funksiyaları mikroprosessorun növündən asılı olur: bəzi mikroprosessorlar çoxlu (müxtəlif) əməliyyatları, digərləri isə məhdud sayda əməliyyatları yerinə yetirə bilirlər.

HMQ-nun funksiyaları ümumi şəkildə mikroprosessorun arxitekturasını müəyyən edir. Mikroprosessorların çoxunda HMQ-su toplama, çıxma, “VƏ”, ”VƏ YA”, “VƏ YA YOX”, inkarətmə, sağa və sola sürüşdürmə, inkrement və dekrement əməliyyatlarını yerinə yetirir.

Bildiyimiz kimi registrlərin bəziləri **xüsusi təyinatlı**, digərləri isə **universal** olurlar. Universal registrlərə **ümumi təyinatlı registrlər (ÜTR)** deyilir. Registrlərin sayı və təyinatı mikroprosessorun arxitekturasından asılı olur.

Mikroprosessorun verilən məlumatlar üzərində müxtəlif əməliyyatlar aparmasında əsas rol oynayan akkumulyatordur. Hesab və məntiq əməliyyatlarının əksəriyyəti HMQ-nın və akkumulyatorun iştirakı ilə yerinə yetirilir. Hər hansı söz (operand) üzərində əməliyyat apararkən onların biri mütləq akkumulyatorda, digəri isə yaddaşda və ya registrlərin birində yerləşdirilir. Əməliyyatın nəticəsi də sonrakı akkumulyatora (operandın yerinə) verilir. HMQ-nın apardığı əməliyyatın nəticəsi də adətən akkumulyatorda yerləşdirilir və bu zaman buna qədər orada olan məlumat itir.

Mikroprosessor bəzi əməliyyatları bilavasitə akkumulyatorda apara bilər. Məsələn, bütün mərtəbələrə 1 yazmaq, akkumulyatorun tərkibini (oradakı ikilik məlumatı) sağa və sola sürüşdürmək, onu inversləmək və s. bir çox əməliyyatları yerinə yetirmək mümkündür.

Akkumulyator məlumatı daxili verilənlər şinindən alır və özündəki məlumatı bu şinə verə bilər. Akkumulyatorun mərtəbələrinin sayı mikroprosessorunda sözün uzunluğuna uyğun olur. Bəzi mikroprosessorlarda sözün uzunluğu ikiqat olur. Məsələn, iki 8-bitlik sözün biri-birinə vurulması zaman alınan 16-bitli nəticə akkumulyatorda ikiqat uzunluqda yazılır.

Digər vacib registr **əmlər sayğacıdır**. Həll olunan hər bir məsələnin proqramı əmlərin ardıcılığından ibarətdir. Əmlər yaddaşda saxlanılır və proqramın icrası zamanı əmlər bir-birinin ardınca yaddaşdan götürülür. Əmlər bir-birinin ardınca

icra olduğundan əmrlər sayğaca hər dəfə yaddaşda yazılmış növbəti əmrin ünvanını hasil edir (formalaşdırır).

Mikroprosessorun işinin əvvəlində ilkin vəziyyətə gətirmə əmrinin köməyi ilə yaddaşın müəyyən bir sahəsindən məlumat əmrlər sayğacına yazılır. Proqramın icrası başlananda sayğacda məhz bu , əvvəlcədən müəyyən edilmiş ünvan yerləşir.

Proqramın icrasından əvvəl proqramın 1-ci əmrinin ünvanı sayğaca yazılmalıdır. 1-ci əmrinin ünvan şini ilə yaddaşa idarə edən sxemə göndərilir və bu ünvana görə yaddaş yuvasının tərkibi oxunur. Sonra bu əmr xüsusi **əmrlər registrinə** verilir.

Əmr yaddaşdan oxunandan sonra mikroprosessor avtomatik olaraq əmrlər sayğacının tərkibinə artım verir. Sayğac bu artımı yaddaşdan indicə oxunan əmrin icrası başlanan andan alır və bu andan etibarən sayğacda növbəti əmrin ünvanı yazılır.

Xüsusi əmrlər qrupu icra edilərkən əmrlər sayğacına başqa məlumat yazıla bilər. Elə ola bilər ki, əsas proqramın əmrlər ardıcılığından başqa budaqlanan proqramın əmrlərini icra etmək tələb olunsun. Bu hissəni hər dəfə proqramda yazmamaq üçün bir dəfə yazılır və hər dəfə mikroprosessor əmrlər ardıcılığından kənara çıxaraq bu hissəni icra edib yenidən geri qayıdır. Proqramın belə ayrıca yazılan hissəsinə **altproqram** deyilir. Belə halda əmrlər sayğacında altproqrama müraciət etmək üçün tələb olunan ünvan yazılır. 8 mərtəbəli mikroprosessorların çoxunda əmrlər sayı 16 mərtəbəli olur. Hər bir əmr deşifrə olunub icra edilənə qədər əmrlər registrində yerləşir. Əmrlər icra üçün ardıcıl seçilərkən yaddaşdan bu registrə giriş məlumatı (əmr) daxil olur.

Yaddaşa hər dəfə müraciət etdikdə mikroprosessorun istifadə edəcəyi yaddaş sahəsi (ikilik ədəd) yaddaşın ünvan registrində saxlanılır. Bu registrin çıxışı **ünvan şini** adlanılır və o yaddaş sahəsini və ya daxil etmə-xaric etmə qurğularını seçmək üçün istifadə olunur.

Yaddaşdan əmr seçilən müddətdə yaddaşın ünvan registri və əmrlər registrinin tərkibi eyni olur, başqa sözlə, yaddaşın ünvan registri yaddaşdan götürülən əmrin yerini göstərir. Əmrin kodu açıldıqdan sonra sayğac artım alır və onun tərkibi yaddaşın ünvan registrindən fərqlənir.

Əmr icra edilən müddətdə yaddaşın ünvanı registrin tərkibi icra edilən əmrdən asılı olur. Əgər əmrə əsasən mikroprosessor yaddaşa heç olmasa bir dəfə də müraciət etməli olsa, onda yaddaşın ünvan registri bu əmrin icrasında ikinci dəfə istifadə olunmalıdır. Bəzi əmrlər üçün, məsələn, akkumulyatoru təmizləmək üçün



yaddaşa ünvanlaşdırma (müraciət) tələb olunmur. Belə əməllərin icrasında yaddaşın ünvan registri yalnız bir dəfə əmri yaddaşdan seçib götürəndə istifadə olunur.

Mikroprosessorun çoxunda yaddaşın ünvan registri və əməllər sayğacının mərtəbələrinin sayı eyni olur. Yaddaşın ünvan registri də sayğac kimi yaddaşın istənilən sahəsinə ünvanlaşdırmaq üçün kifayət edən mərtəbələr sayına malik olmalıdır. 8 mərtəbəli mikroprosessorların əksəriyyətində ünvanın yaddaş registrinin mərtəbələr sayı 16-dır.

Yaddaşın ünvan registri həm də daxili məlumat şininə qoşulduqdan, o müxtəlif mərtəbələrdən yüklənə bilər. Mikroprosessorların çoxu bu registrə sayğacın, ÜTR-in və ya yaddaşın hər hansı bir hissəsinin tərkibini yazmaq üçün əməllərə malik olurlar. Əməllərin bəziləri yaddaşın ünvan registrinin tərkibini müəyyən hesablamalar yerinə yetirməklə dəyişməyə imkan verirlər: registrin yeni tərkibi əməllər sayğacının tərkibinə əməldən göstərilən ədədi əlavə etməklə və ya onu çıxmaqla alınır. Belə ünvanlaşdırma **sürüşdürülməli ünvanlaşdırma** adlanır.

Bufer (aralıq) registri məlumatı müvəqqəti saxlamaq (buferləşdirmək) üçün istifadə olunur.

**Əlamət (vəziyyət) registri.** Bu 5 mərtəbəli registr proqramın icrası zamanı bir çox əməliyyatların yerinə yetirilməsi nəticəsində alınan xüsusi əlamətlərin saxlanması üçündür. Proqramda keçid olanda verilən əlamətə görə əməlin icrası yaddaşın hər hansı bir yeni sahəsindən başlayır, yəni əməllər sayğacına yeni ədəd yüklənir. Şərti keçid vəziyyətində belə yüklənmə o vaxt baş verir ki, müəyyən yoxlamaların nəticəsi gözlənilən qiymətlərə uyğun gəlir. Bu nəticələr vəziyyət registrində saxlanılır. Vəziyyət registri proqramçıya mikroprosessorun işini əməllərin icra qaydasını dəyişməklə təşkil etməyə imkan verir. Bu registrin 5 triggeri aşağıdakı kimi təyinatlanır:

1. **Ötürmə-keçid triggeri** sonuncu icra olunan əməliyyatın ötürmə ilə və ya borc götürmə (mənfi ötürmə) ilə müşayiyyət edildiyini göstərir. İki ədədin toplanması nəticəsində HMQ-nun ən yüksək mərtəbəsindən ötürmə olarsa ötürmə mərtəbəsinə 1 yazılır. Böyük ədədi kiçik ədəddən çıxarkən registrdə mənfi ötürmə (borc götürmə) qeyd edilir.

2. **Sıfır nəticəsi triggeri.** Əməliyyat gurtarandan sonra registrin bütün mərtəbələrində 0 alınarsa, bu mərtəbə 1 qiymətini alır.

3. **İşarə (əlamət) triggeri.** HMQ-da yerinə yetirilən əməliyyatın nəticəsinin böyük mərtəbəsinə uyğun vəziyyətə keçir.

4. **Cütlük registri-** mərtəbələrdə 1-lərin sayı cüt olarsa 1 vəziyyətinə keçir.

**5. Əlavə keçid registri.** Əməliyyat yerinə yetirilən zaman 4-cü mərtəbədən alınan keçid mərtəbəsini yadda saxlayır.

Mikrorprocessorların əksəriyyətinin tərkibində yaddaş qurğuları kimi istifadə olunan registrlər yığımı – ÜTR-lər olur. HMQ ÜTR-nin tərkibi ilə (xarici ünvan və məlumat şininə çıxmadan) bilavasitə əməliyyat apara bildiyindən əməliyyatların icra sürəti daha yüksək olur. Ona görə ÜTR-lərə əksər hallarda **ifrat operativ yaddaş** deyilir.

**Stekin** əsas vəzifəsi prosessorun registrlərinin tərkibindəkilərini yadda saxlamaqdır. Stek üçün YSYQ- da sahə ayrılır. Stek yüzrlərlə ardıcıl bitlərdən ibarətdir və yaddaş kartında ünvanı vardır.

Prossessoru çox vacib stekin ilkin ünvanıdır, belə ki, qalan baytları ünvanlaşdırmaq olar. Bu məqsədlə prosessoru xüsusi 16 bitli registr vardır və stek göstəricisi adlanır. Stekin YSYQ sahəsində ilkin ünvanı vardır və arakəsmə yarandıqda prosessorun bütün registrlərinin tərkibindəkiləri yadda saxlamaq üçün təyin edilmişdir.

Stek sahəsi onluq ünvanları 256-511 tutur. Stekin birinci baytı 511 ünvanında, axırını 256 ünvanında yerləşir.

Göstərici sözü ünvan deməkdir. 16 bitli stek göstəricisində ünvan bitləri stekin birinci baytını göstərir. Stek yaddaşın istənilən sahəsində (256-511) yerləşə bilər, bu səbəbdən ünvanın yadda saxlanması üçün 16 bitli registr tələb olunur. Hər bir oyma 8 bitli registrdir. Arakəsmə daxil olduqda prosessorun bütün registrlərinin stek göstəricisindən başqa, tərkiblərini stekdə qorumaq olar. 8 bitli registr üçün stekin bir baytı kifayətdir, 16 bitli registr üçün isə stekin iki baytı istifadə olunur. Stek göstəricisinin tərkibini yadda saxlamaq tələb olunmur, onun vəzifəsi stekin ilkin ünvanını yadda saxlamaqdır.

**Kəsilmə** deyəndə cari proqramın icrasının müvəqqəti dayandırılması və kəsilməsini yaradan qurğuya xidmət göstərən proqramın icrasına keçid nəzərdə tutulur. Kəsilmə bir növ digər qurğulardan (yaddaşdan, daxil etmə-xaric etmə qurğulardan və s.) idarəetmə sxemlərinə gələn sorğudur. Kəsilmə üçün sorğu gələndə idarəetmə sxemləri digər qurğuların nə vaxt və hansı ardıcılıqla daxili məlumat şinindən istifadə etməsi haqqında qərar qəbul edir.

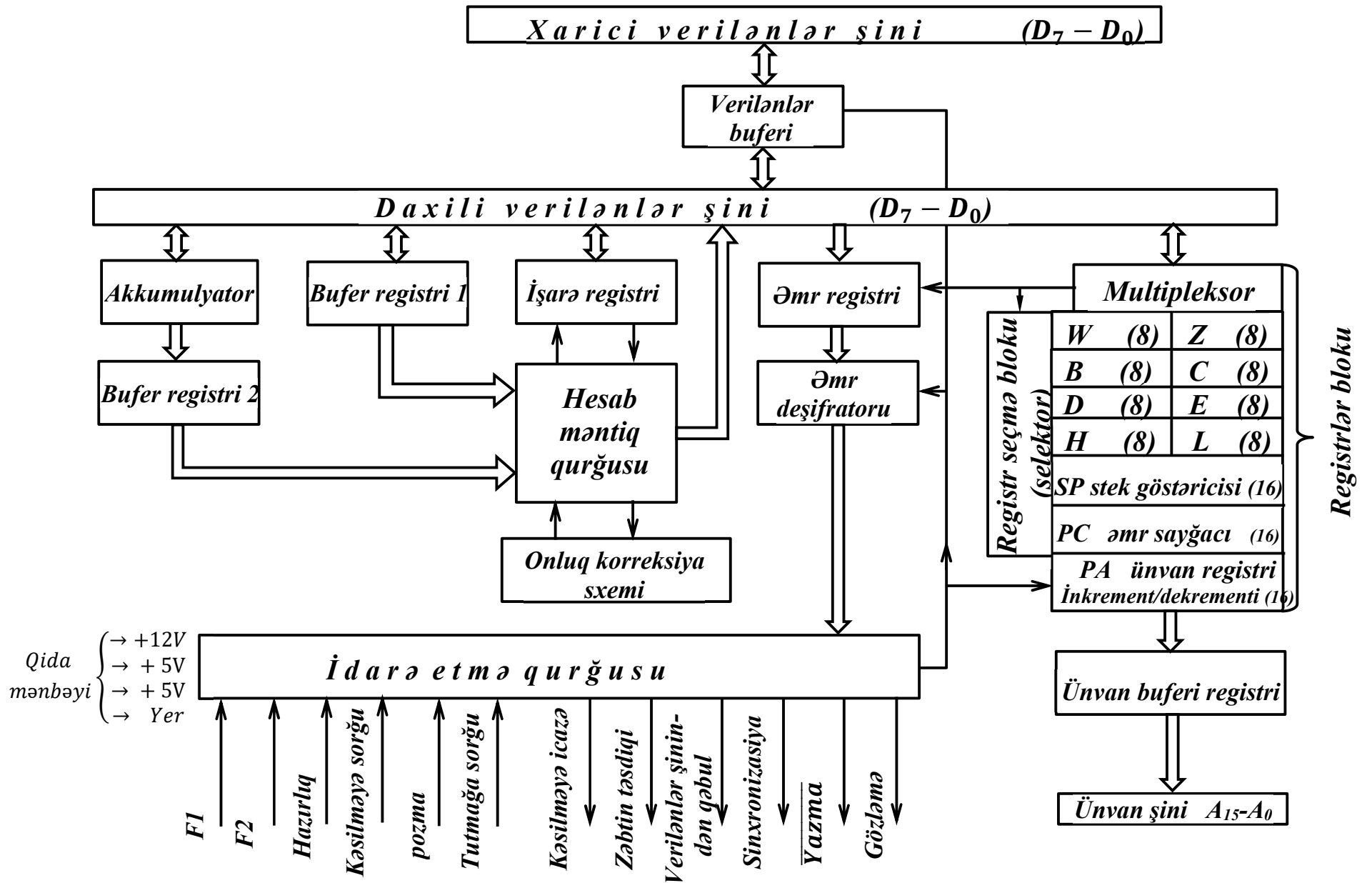
**İdarəetmə qurğusu.** alınmış əmrə uyğun idarəetmə signalı formalaşdırır. MP-nin idarəetmə hissəsinə əmr registri, deşifrator və proqram saygacı daxildir.

**Ümumi vəzifəli registrlər (ÜVR)** – (B, C, D, E, H, L) verilənlərin müvəqqəti saxlanması, və 16 mərtəbəli verilənlər (operanda) ünvanının göstəricisi vəzifəsini y/y.

**Onluq korreksiya sxemi** xüsusi əmrlə qoşulur və ikilik hesablamadan istifadə edən HMQ-na hesabi əməllərin ikilik – onluq kolaşdırmaya uyğun cəmləməni həyata keçirməsinə imkan verir.

**Bufər (aralıq) registrləri.** Verilənlər buferi və ünvan buferi mərkəzi prosessorla xarici verilənlər və ünvan şini arasında əlaqəni təmin etmək üçündür. Buferin əsas xüsusiyyəti ondan ibarətdir ki, o hər mərtəbədə (dərəcədə) 3 vəziyyətli məntiq elementlərindən istifadə edir. Bu elementlərin 0 və 1 vəziyyətlərindən başqa bir vəziyyəti də nəzərə alınır ki, onlar paktiki olaraq sonsuz çıxış gərginliyinə malik olurlar. Belə buferlər mikroprosessorla xarici şindən (verilənlər və ünvan) ayrılmağa imkan verməklə eyni bir verilənlər şini ilə həm verilənlərin qəbul olunmasını həm də xaric olunmasına imkan yaradır və bununla da mikrosxemin çıxışlarının sayını azaltmış olur.

Mikrokontroller sisteminin tipik idarəetmə sisteminin strukturu



## Mikrokontrollerin tipik struktur sxemi

Mikrokontroller, müxtəlif funksiyaları yerinə yetirmək üçün nəzərdə tutulmuş mini-kompüter olan bir mikrosxemdir. Başqa cür desək, Mikrokontroller müxtəlif elektron cihazları idarə etmək üçün nəzərdə tutulmuş xüsusi mikrosxemdir. Mikrokontrollerlər ilk dəfə ümumi təyinatlı mikroprosessorlarla eyni ildə meydana çıxdı (1971). Bu çip proqramçı tərəfindən yaradılmış proqrama uyğun işləyir. Mikrokontrollerlərin tərtibatçıları dahiyənə bir fikir irəli sürdülər - prosessoru, yaddaşı, ROM-u və periferiyaları adi mikrosxem kimi görünən bir paket daxilində birləşdirmək. O vaxtdan bəri mikrokontrollerlərin istehsalı hər il prosessorların istehsalını dəfələrlə üstələyir və onlara tələbat azalmayıb. Mikrokontroller onun imkanlarını, eləcə də qiymətini müəyyən edən müxtəlif sayda periferik modullardan ibarət ola bilər. Mikrokontrollerin periferik hissələrinə aiddir: ADC (analoqdan rəqəmə çevirici), müxtəlif taymerlər, analoq komparator, UART (Universal asinxron qəbuledici-ötürücü), USB, CAN və s. Lakin, bir qayda olaraq, hər hansı bir mikrokontroller aşağıdakı əsas qovşaqları əhatə edir: Hesab məntiq qurğusu MQ, Operativ yaddaş, Sabit yaddaş, Takt tezlik generatoru, Giriş çıxış qurğusu, Taymerlər. Mikrokontrollerin sadələşdirilmiş sxemi aşağıdakı kimidir:

1. **Serial Peripheral Interface, SPI** - Ardıcıl əlaqə interfeysi (3 xətlə)
2. **FLASH** - Proqramın yaddaşda saxlanması üçün yenidən proqramlaşdırıla bilən yaddaş
3. **EEPROM** (Electrically Erasable Programmable Read-Only Memory) - Yenidən proqramlaşdırıla bilən sabit yaddaş qurğusu (YSYQ), çərəyandan asılı olmayan yaddaş
4. **CPU** (ЦП) - Mərkəzi idarəetmə prosessoru, Mikrokontrollerin ürəyi, 8 bitli mikroprosessor nüvəsi,
5. **Program Counter** - əmirlər sayğacı
6. **Instruction Register** - Əmr registri
7. **32 General Purpose Registers** – 32 Ümumi təyinatlı registr
8. **RAM** (Random Access Memory) - prosessorun operativ yaddaşı
9. **ALU** (АЛУ) -Hesab məntiq qurğusu
10. **Watchdog Timer** - Qoruyucu və ya nəzarət taymeri
11. **I/O Ports** - Daxiletmə-xaricətmə portu
12. **Interrupts** - İdarəetmə bloku və kəsilməyə reaksiya
13. **Timers/Counters** - Taymer və sayğac modulları
14. **LCD Interface** (Liquid-Crystal Display Interface) Maye-kristal displeyin, indikatorun qoşulması üçün interfeys
15. **Analog Comparator** - analoq komparatoru, analoq siqnallarının müqayisə bloku
16. **A/D Converter** (Analog/Digital converter) - Analoq rəqəm çeviricisi

1. **Serial Peripheral Interface, SPI** - Ardıcıl əlaqə interfeysi SPI - bir neçə mikrokontroller arasında bir neçə MHz (saniyədə bir neçə million takt) sürəti ilə verilənlərin mübadiləsini həyata keçirir. 2 qurğu arasında SPI interfeysi vasitəsi ilə verilənlərin mübadiləsi üçün 3 xətt bəs edir:

1. MOSI (Master Output Slave Input) – İdarə edəndən idarə olunana oturulən məlumat (Mənbədən təyinatla verilənlərin ötürülməsi).

2. MISO (Master Input Slave Output) – İdarə olunandan idarə edənə oturulən məlumat (Təyinatdan mənbəyə verilənlərin ötürülməsi).

3. CLK (Clock) - Takt siqnalı.

**SPI** interfeysli qurğular 2 yerə bölünür: Mənbə və təyinat. Əyər interfeysə bir neçə qurğu qoşularsa, bu qurğular arasında verilənlərin mübadiləsi üçün əlavə mübadilə xətləri lazımdır ki, mənbə təyinat qurğusunu taparaq ona sorğu göndərsin. Həmçinin SPI interfeysi sxem daxili SPI proqramlaşdırılması üçün istifadə olunur. Proqramlaşdırıcı mikrokontrollərə bu interfeyslə qoşulur.

2. **FLASH- yaddaş proqramı**- flash texnologiyasıyla hazırlanmış, sabit yaddaş qurğusudur. Burada mikrokontrollerin HMQ-da istifadə olunacaq proqramlar saxlanılır. Yerine yetirmək üçün proqramın kodunu dəyişərək və əlavələr edərək, Flash yaddaş çipinə dəfələrlə yenidən yazmaq olur. Bu tip yaddaş ona yazılmış verilənləri 40 il saxlaya bilər və mümkün olan silmə/yazma dövrlərinin sayı 10 000-ə qədərdir. Mikrokontrollerin modelindən asılı olaraq flash yaddaşın həcmi 256KB-ə qədərdir.

3. **EEPROM – (YSYQ) Mikrokontrolleri qida mənbəyindən ayırdıqda belə verilənləri yadda saxlayan yaddaşdır.** Bu yaddaşda yerinə yetirilən proqramların parametrlərini, qurğunun işi haqqında yığılmış məlumatları və digər lazımlı informasiyaları saxlamaq olur. Məsələn: mikrokontrollerdə kiçik metroloji stansiya yığıb, YSYQ də hər gün havanın temperaturu, təzyiqi, küləyin gücü haqqında verilənləri saxlayaraq və sonra istənilən anda bu toplanmış informasiya ilə statik analizlər aparmaq olur.

**EEPROM üçün, Ram və FLASH -in ünvan fəzasından fərqlənən ünvan fəzası ayrılıb. Mikrokontroller üçün EEPROM yaddaşı çox vacib mənbədir. Çünki onun həcmi çox azdır - 1 çip üçün 0.5-dən bir neçə KB-ə qədər. Bu tip yaddaş üçün yenidən yazma miqdarı 100000-dir. Bu FLASH yaddaşın tutumundan 10 dəfə çoxdur.**

4. **CPU** – Mərkəzi idarəetmə prosessoru- Mikrokontrollerin beyni olmaqla, HMQ, registrlər və operativ yaddaşdan ibarətdir. HMQ-na 32 ümumi təyinatlı registrlər bloku qoşulmuşdur ki, onların da hər biri 1 bayt yaddaşdan (8 bit) ibarətdir. Ümumi təyinatlı registrlərin ünvan sahəsi operativ yaddaşın (RAM) başlanğıcında yerləşir, lakin onun bir hissəsi deyil. Registrlərə yerləşdirilən verilənlərlə müxtəlif hesabi, məntiqi və digər əməliyyatlarını yerinə yetirmək olur. Operativ yaddaşda (RAM-da) belə əməliyyatları yerinə yetirmək mümkün deyil. Operativ yaddaşdan verilənlərlə işləmək üçün onları registrlərə yazmaq,

registrlərdə lazımi əməliyyatları yerinə yetirmək, sonra isə əldə olunan məlumatları registrlərdən yaddaşa və ya başqa registrlərə yazaraq hansısa hərəkəti yerinə yetirmək lazımdır.

Yuxarıda qeyd etdiyimiz kimi, HMQ-u rəqəmlər üzərində əməliyyatları yerinə yetirir və əməliyyatın nəticəni rəqəm halında qaytarır. Alınan nəticə **ümumi təyinətli registrlərdə**- xüsusi, müvəqqəti yaddaşda saxlanılır. Hər bir mikrokontrollerin registrlərinin sayı müxtəlif ola bilər. Yuxarıda göstərdiyimiz mikrokontrollerin struktur sxemində 32 **ümumi təyinətli registr** vardır. Buna baxmayaraq mikrokontrollerin yaxşı işləməsi üçün bu qədər ümumi təyinətli registrlər azdır. Belə ki, 32 bayt çox kiçik yaddaş tutumudur. Daha çox məlumat yadda saxlamaq üçün operativ yaddaş qurğusundan (OYQ) istifadə edilir. ÜTR-lər HMQ-nun hal hazırda emal etdiyi verilənləri saxlayır, OYQ-su isə digərlərini.

**HMQ**-nun yerinə yetirdiyi əmrlər, daha dəqiq desək, ardıcıl əmrlər sabit yaddaş qurğusunda (SYQ) saxlanılır. Bu əsasən **Flash- yaddaşdır**. Verilmiş əmrlər ardıcılığı mikrokontroller üçün proqrammist tərəfindən yaradılmış proqramdır. Bütün əmrlər SYQ-də müəyyən ünvanlarda yerləşir.

Hansısa əmiri SYQ-dən götürmək üçün onun ünvanına müraciət etmək lazımdır. Bununla **proqram sayğacı** və ya **əmlər sayğacı** məşğul olur.

SYQ-dən verilənlər **əmr registrinə** daxil olur. HMQ-su həmişə **əmr registrinin** məzmununa \* baxır\* və orada əmr olarsa HMQ həmin anda onu yerinə yetirməyə başlayır.

6. **RAM** - operativ yaddaş. O, registrlərdən verilənləri yazsa, məlumatları registrlərə oxuya bilər. Verilənlərlə bütün əməliyyatlar və hesablamalar registrlərdə yerinə yetirilir.

7. **Hesab Məntiq Qurğusu** (HMQ) mikrokontrollerin ürəyi hesab olunur. HMQ ikili verilənlər üzərində hesabi və məntiqi əməliyyatları yerinə yetirir. HMQ-su 8-, 16- və ya 32-bitli olurlar. Məsələn, HMQ 8 bitdirsə, o zaman iki 8 bitlik rəqəm üzərində əməliyyat həyata keçirə və əməliyyatın 8 bitlik nəticəsini yarada bilər. Hesabi əməliyyatlara aşağıdakılar daxildir: toplama, çıxma, müqayisə və s. **Məntiqi əməliyyatlara** aşağıdakılar daxildir: vurma "VƏ", toplama "VƏ YA", inkar "NOT", "VƏ YA YOX", sağa sürüşdürmə, sola sürüşdürmə və s. Bunlardan əlavə nə məntiqi, nə də hesabi əməliyyatlara aid olmayan "0"-a sıfırlamaq və ya "1"-ə təyin etmək kimi əməliyyatlar da var.

Bildiyiniz kimi, hər bir mikrokontrollerin **ümumi təyinətli registrləri** vardır ki, onlar hesab məntiq qurğusu (HMQ) tərəfindən emal olunmuş rəqəmləri saxlayan operativ yaddaş kimi fəaliyyət göstərirlər. Üstəlik, proqramın icrası zamanı bu

rəqəmlər dəyişə bilər: artar, azalar, sıfırlanar, yenidən yüklənər və s. Yaddaş yuvasının ölçüsü prosessorun dərəcəliyindən asılıdır: 8, 16 və ya 32 bit. Hər bir yaddaş yuvasının öz ünvanı var və bu ünvanı istifadə edərək ona rəqəm yazmaq olar.

Ümumi təyinatlı registrlərlə yanaşı, mikrokontrollerdə çox mühüm rol oynayan **xüsusi təyinatlı registrlər** də var. Onlar olmadan mikrokontroller mənasız bir \*dəmir\* parçası olardı. **Xüsusi təyinatlı registrlər** mikrokontrollerin əsas və periferik modullarının (taymerlər, ADC-lər, analoq komparatorlar, UART, USB və s.), həmçinin giriş/çıxış portlarının funksiyalarını idarə edir. Ümumi təyinatlı registrlərdən fərqli olaraq, xüsusi təyinatlı registrlərin ciddi şəkildə öz ünvanları vardır. **Xüsusi təyinatlı registrlərin** təyinatını dəyişdirməklə mikrokontrollerin bütün modullarının işini idarət etmək olar.

**Qoruyucu və ya nəzarət taymeri (Watchdog Timer)** cihazın donması ilə sonradan yenidən işə salınmasını izləmək üçün bir sistemdir. Bu, səhv əməliyyat sistemi olan köhnə kompüter üçün avtomatik RESET düyməsinə bənzəyir.

**Giriş/Çıxış Portları, GPIO** giriş/çıxış portlarının blokları toplusudur. Mikrokontrollerin bütün bu qurğuları **giriş - çıxış** portları (şinləri) olmadan heç bir işə yaramazdı. Bu portlar vasitəsilə mikrokontroller xarici aləmlə qarşılıqlı əlaqəni təmin edir. **Giriş- çıxış portlarını** elə sazlamaq olar ki, onlar həm giriş, həm də çıxış kimi işləyə bilsinlər. Bu portların idarə olunması xüsusi registrlərin vasitəsilə həyata keçirilir.

**Kəsilmələr** - Hər kəsilmə üçün ayrıca alt proqram hazırlanır və yaddaşa yazılır.

Niyə bu blok kəsilmə bloku adlanır? - çünki kəsilmə üçün müəyyən edilmiş hadisə baş verdikdə əsas proqramın icrası dayandırılır və cari kəsilmə üçün yazılmış alt proqramın prioritet icrası baş verir. Alt proqramın icrası başa çatdıqdan sonra əsas proqram kəsildiyi nöqtəyə qayıdır.

**Taymers/Counters** - taymerlər və sayğaclar yığımıdır. Mikrokontroller, bir qayda olaraq, birdən dördə qədər taymer və sayğacdən ibarətdir. Onlar xarici hadisələrin sayını hesablamaq, müəyyən uzunluqlu (müddətli) siqnallar yaratmaq, kəsilmə sorğuları yaratmaq və s. üçün istifadə edilir. Taymerlər və sayğaclar 8 və 16 bitli olurlar.

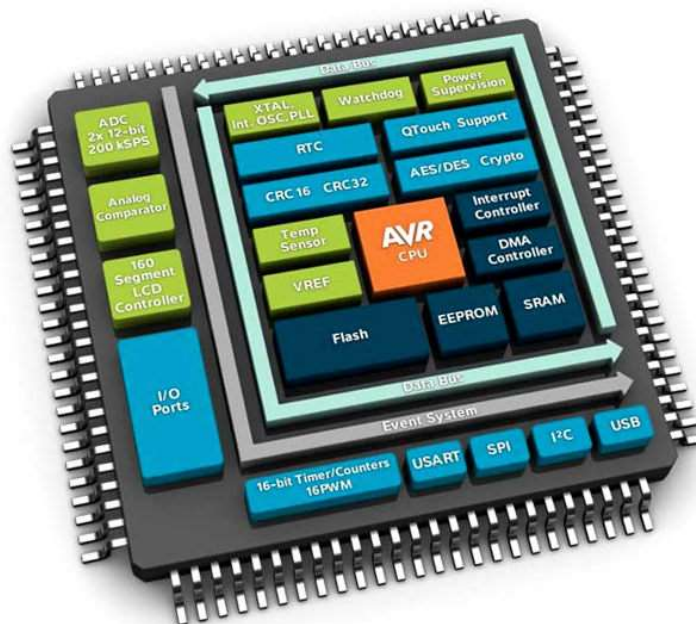
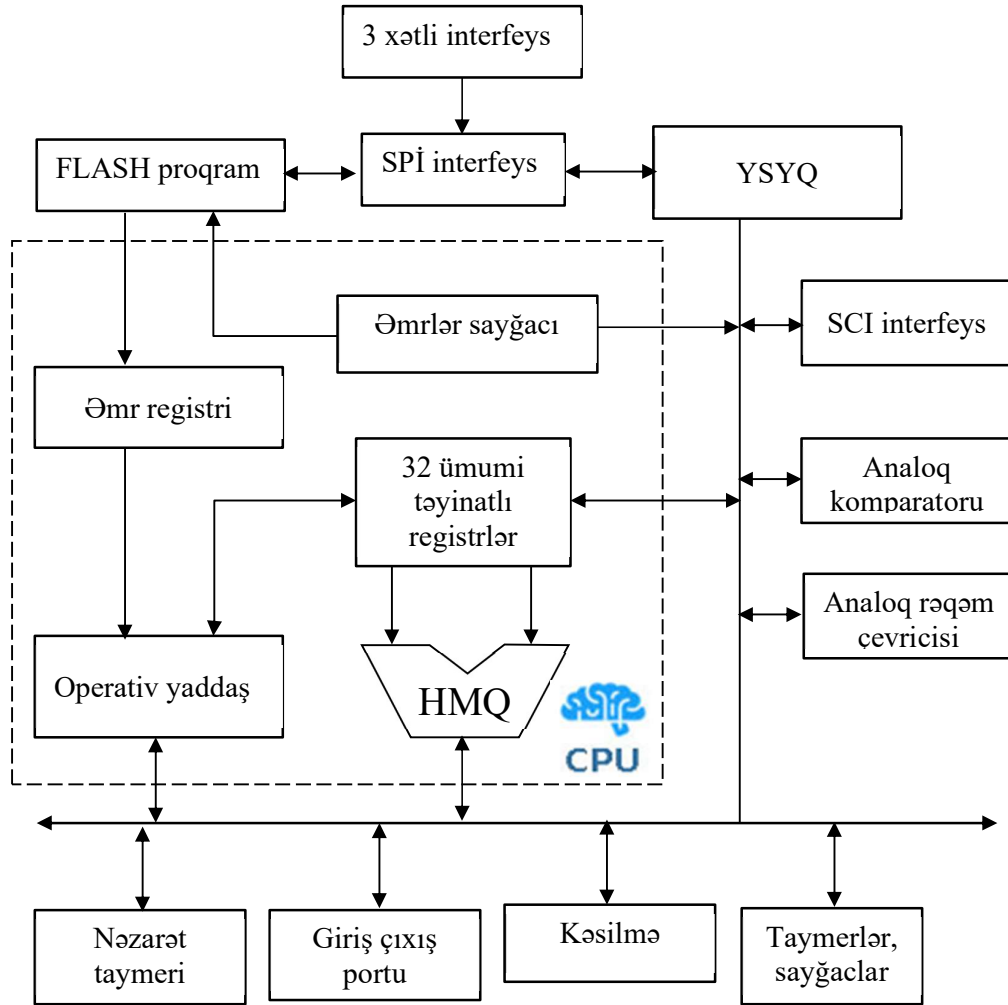
Onu da qeyd edək ki, mikrokontrollerin bütün işi **Takt tezliyi generatoru** vasitəsilə sinxronlaşdırılır ki, o da daxili və xarici ola bilər.

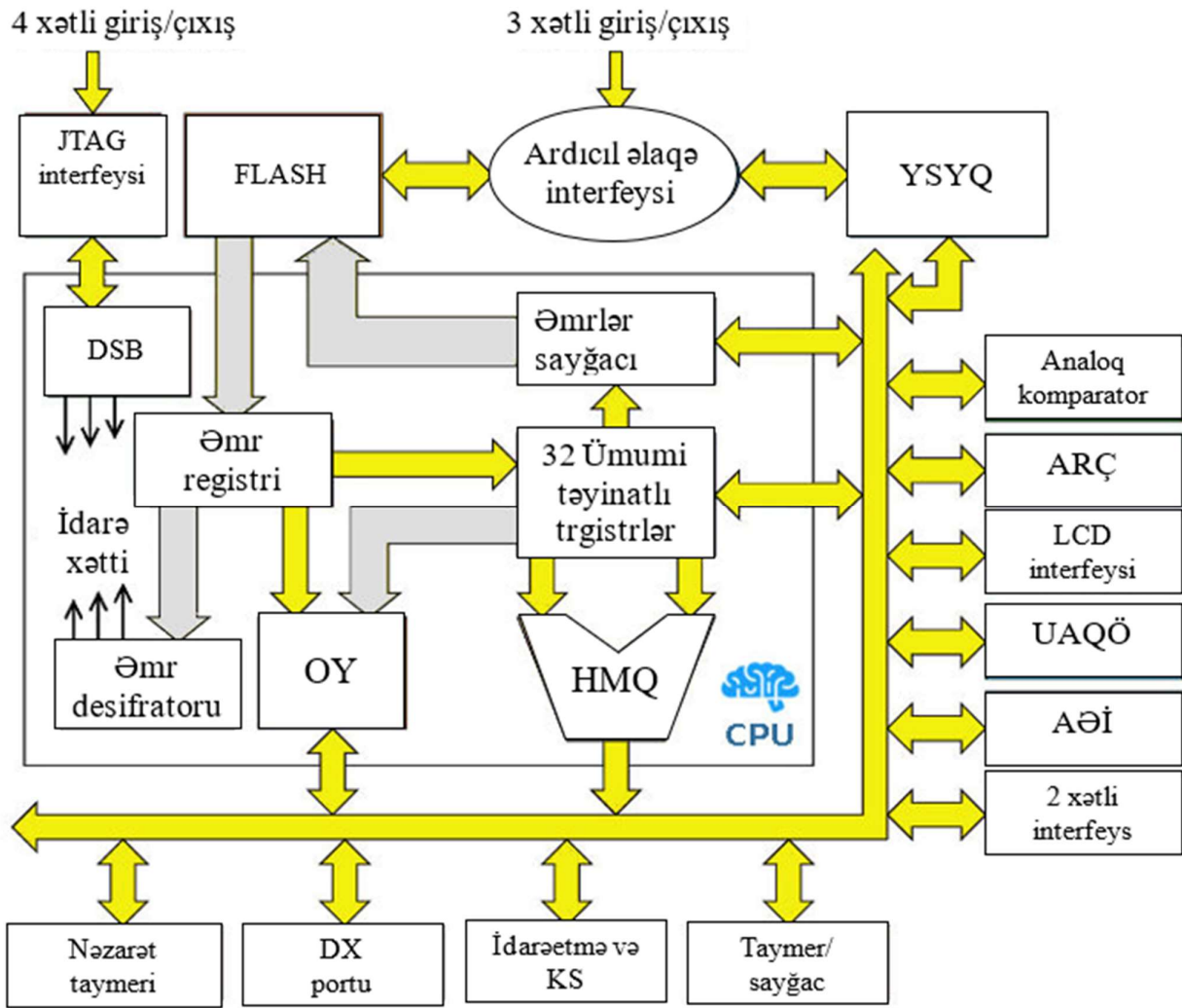
**LCD Interface** (Liquid-Crystal Display Interface) Maye-kristal displeyin, indikatorun qoşulması üçün interfeys.



**Analog Comparator** - analoq komparatoru, analoq siqnallarının müqayisə bloku.

**A/D Converter (Analog/Digital converter)** - Analoq rəqəm çevricisi





**JTAG interfeysi**- daxili sazlama blokundan (DSB) istifadə edərək (emulyatordan istifadə etmədən), birbaşa cip üzərində daxili sazlamani heyata keçirən vacib interfeysdir. Demək olar ki, **JTAG** - mikrokontrollerin “dəmir” sazlama interfeysidir. Mikrosxem, programlaşdırma və sazlamaniin program kompleksinə, JTAG adapteri vasitəsiylə birbaşa qoşulur.

Bu interfeysi istifadə etməklə, programi addim rejimində birbaşa mikrokontrollerdə yerinə yetirmək, mikrokontrollerə qoşulmuş registrlərin məzmununun dəyişməsini, indikatorların və işıq diodlarının hər addimdan sonra yanib sönməsini izlemek olar və s. **JTAG interfeysinə** qoşulmaq ucun 4 xətlı interfeys bəs edir: TDI (verilenler girişı), TDO (verilenler çıxışı),TCK (tezlik), TMS (rejim seçimi)

**FLASH- yaddash proqrami**- flash texnologiyasiyla hazirlanmış, sabit yaddaş qurgusudur. Burada mikrokontrollerin HMQ-da istifadə olunacaq programlar saxlanilir. Yerine yetirmek ucun programin kodunu deyişərək və əlavələr edərək, Flash yaddaş çipine dəfələrlə yenidən yazmaq olur. Bu tip yaddaş

ona yazılmış verilənləri 40 il saxliya bilər və mümkün olan silmə/yazma dövrlərinin sayı 10 000 cata bilər. Mikrokontrollerin modelindən aslı olaraq flash yaddaşın həcmi 256KB cata bilər.

**Serial Peripheral Interface, SPI** - Ardıcıl əlaqə interfeysi SPI - bir neçə mikrokontroller arasında bir neçə MHz (saniyədə bir neçə million takt) sürəti ilə verilənlərin mübadiləsini həyata keçirir. 2 qurğu arasında SPI interfeysi vasitəsi ilə verilənlərin mübadiləsi üçün 3 xətt bəs edir:

1. MOSI (Master Output Slave Input) – İdarə edəndən idarə olunana oturulən məlumat (Mənbədən təyinat verilənlərin ötürülməsi).

2. MISO (Master Input Slave Output) – İdarə olunan idarə edənə oturulən məlumat (Təyinatdan mənbəyə verilənlərin ötürülməsi).

3. CLK (Clock) - Takt siqnalı.

**SPI** interfeysli qurğular 2 yerə bölünür: Mənbə və təyinat. Əyər interfeysə bir neçə qurğu qoşularsa, bu qurğular arasında verilənlərin mübadiləsi üçün əlavə mübadilə xətləri lazımdır ki, mənbə təyinat qurğusunu taparaq ona sorğu göndərsin. Həmçinin SPI interfeysi sxem daxili SPI proqramlaşdırılması üçün istifadə olunur. Proqramlaşdırıcı mikrokontrollərə bu interfeyslə qoşulur.

**EEPROM - Mikrokontrolleri qida mənbəyindən ayırdıqda belə verilənləri yadda saxlayan yaddaşdır.** Bu yaddaşda yerinə yetirilən proqramların parametrlərini, qurğunun işi haqqında yığılmış məlumatları və digər lazımlı informasiyalari saxlamaq olar. Məsələn: mikrokontrollerdə kiçik metroloji stansiya yigib, YSYQ də hər gün havanın temperaturu, teyziqi, küləyin gücü haqqında verilənləri saxlayaraq və sonra istənilən anda bu toplanılmış informasiya ilə statik analizlər aparmaq olar.

**EEPROM üçün, Ram və FLASH -in ünvan fəzasından fərqlənən ünvan fəzası ayrılıb. Mikrokontroller üçün EEPROM yaddaşı çox vacib mənbədir. Çünki onun həcmi çox azdır - 1 çip üçün 0.5 dən bir neçə KB-a qədər. Bu tip yaddaş üçün yenidən yazma miqdarı 100000-dir. Bu FLASH yaddaşın tutumundan 10 dəfə çoxdur.**

**HMQ- hesab məntiq qurğusu takt siqnalına sinxron olaraq və əmirlər sayğacının vəziyyətinə əsaslanaraq yaddaşdan (FLASH) növbəti əmri seçir və onu yerinə yetirir.**

**CPU – Mərkəzi idarəetmə prosessoru- Mikrokontrollerin beyni olmaqla, HMQ, registrlər və operativ yaddaşdan ibarətdir. HMQ-na 32 ümumi təyinatlı registrlər bloku qoşulmuşdur ki, onların da hər biri 1 bayt yaddaşdan (8 bit) ibarətdir. Ümumi təyinatlı registrlərin ünvan sahəsi operativ yaddaşının (RAM)**

başlanğıcında yerləşir, lakin onun bir hissəsi deyil. Registrlərə yerləşdirilən verilənlərlə müxtəlif hesabi, məntiqi və digər əməliyyatlarını yerinə yetirmək olar. Operativ yaddaşda (RAM-da) belə əməliyyatları yerinə yetirmək mümkün deyil. Operativ yaddaşdan verilənlərlə işləmək üçün onları registrlərə yazmaq, registrlərdə lazımi əməliyyatları yerinə yetirmək, sonra isə əldə olunan məlumatları registrlərdən yaddaşa və ya başqa registrlərə yazaraq hansısa hərəkəti yerinə yetirmək lazımdır.

**RAM** - operativ yaddaş. O, registrlərdən verilənləri yaza, məlumatları registrlərə oxuya bilir. Verilənlərlə bütün əməliyyatlar və hesablamalar registrlərdə yerinə yetirilir.

**Analog Comparator** - bu blok iki siqnal səviyyəsini müqayisə edir və müqayisə nəticəsini müəyyən registrdə saxlayır, bundan sonra nəticə təhlil oluna və lazımi hərəkətlər yerinə yetirilə bilər. Məsələn: bu blokdan ARC (Analog-to-Digital Converter) kimi istifadə edə və qida gərginliyini ölçə bilərsiniz, əgər qida gərginliyi aşağı səviyyəyə çatmışdırsa - bəzi hərəkətləri yerinə yetirilir, qırmızı işıqdiodları yanıb-sönür və s. Həmçinin, bu moduldan analoq siqnalların müddətini ölçmək, potensiometrədən istifadə edərək cihazın müəyyən edilmiş iş rejimlərini oxumaq və s.

**A/D Converter** - bu blok analoq gərginlik dəyərini proqramda istifadə oluna bilən və onun əsasında müəyyən hərəkətlərin həyata keçirilə biləcəyi rəqəmsal dəyərə çevirir. Bir qayda olaraq, AVR mikrokontrolöründə ARC-nin girişinə tətbiq olunan gərginlik diapazonu 0-5,5 Volt diapazonundadır. Bu qurğu üçün mikrokontrollerin sabit və yüksək keyfiyyətli enerji mənbəyi ilə təchiz edilməsi çox vacibdir. Bir çox AVR mikrokontroller cihazında ARC dövrəsini sabit enerji ilə təmin etmək üçün xüsusi ayrıca çıxış var.

**LCD interfeysi** - mayekristal indiqikatorun və ya ekranı birləşdirmək üçün interfeys. Onlar məlumatı, cihazın vəziyyətini və onun qovşaqlarını göstərmək üçün istifadə olunur.

**USART** digər cihazlarla məlumat mübadiləsi üçün seriyalı ardıcıl asinxron interfeysdir.

**I<sup>2</sup>C** - iki xətlə şinlə məlumat mübadiləsi üçün interfeys. Belə İki məlumat xəttindən istifadə etməklə 128-ə qədər müxtəlif qurğu qoşula bilər: takt siqnalı (SCL) və verilənlər siqnalı (SDA). I<sup>2</sup>C interfeysi I<sup>2</sup>C interfeysinə əsas versiyasının analoqudur.

SPI interfeysindən (bir mənbə və bir / bir neçə qəbuledici) fərqli olaraq, TWI interfeysi iki istiqamətlidir ki, bu da bir neçə mikrokontroller arasında kiçik daxili şəbəkə təşkil etməyə imkan verir.

**Watchdog Timer** cihazın donması ilə sonradan yenidən işə salınmasını izləmək üçün bir sistemdir. Bu, səhv əməliyyat sistemi olan köhnə kompüter üçün avtomatik RESET düyməsinə bənzəyir.

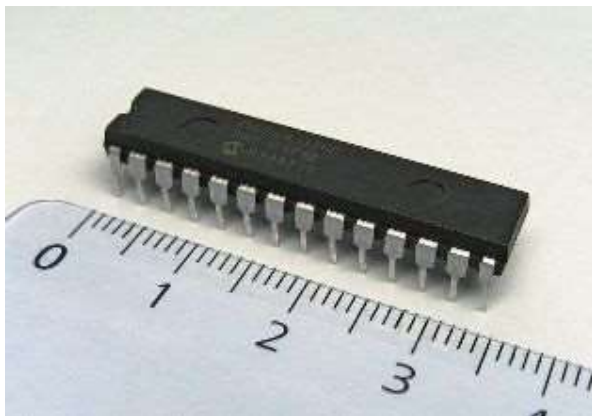
**Giriş/Çıxış Portları, GPIO** giriş/çıxış portlarının blokları toplusudur ki, onlara müxtəlif sensorları, icraçı qurğuları və sxemləri qoşmaq olar. Mikrokontrollerdəki portlardan gələn giriş / çıxış pinlərinin sayı 3 ilə 86 arasında ola bilər.

**Kəsilmələr** - Hər kəsilmə üçün ayrıca alt proqram hazırlanır və yaddaşa yazılır.

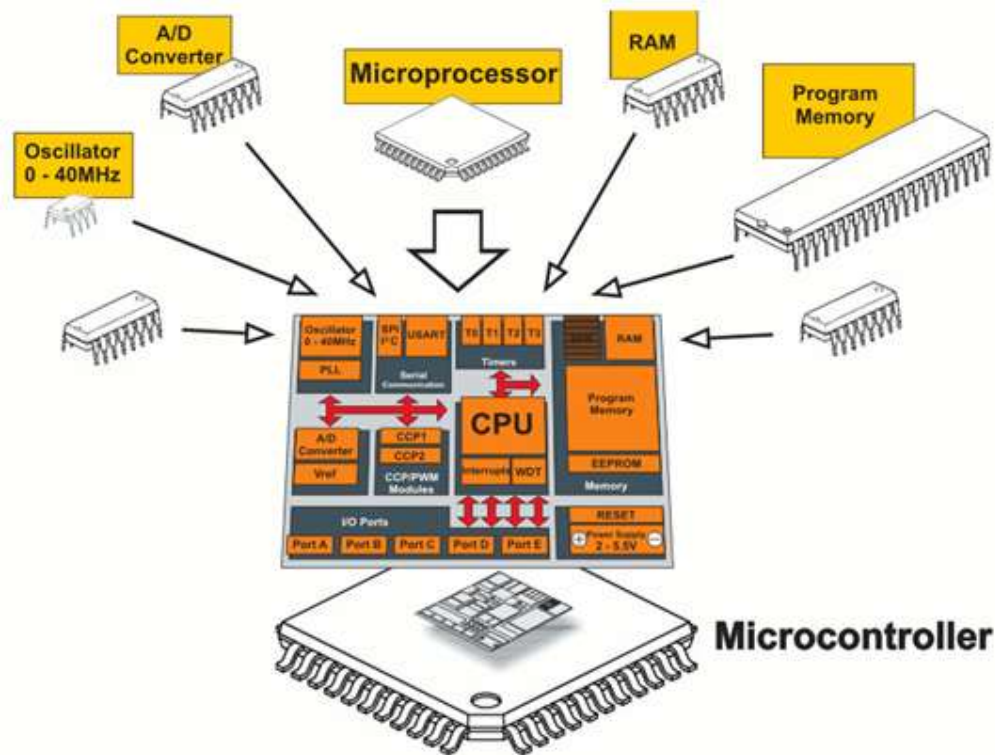
Niyə bu blok kəsilmə bloku adlanır? - çünki kəsilmə üçün müəyyən edilmiş hadisə baş verdikdə əsas proqramın icrası dayandırılır və cari kəsilmə üçün yazılmış alt proqramın prioritet icrası baş verir. Alt proqramın icrası başa çatdıqdan sonra əsas proqram kəsildiyi nöqtəyə qaydır.

**Taymers/Counters** - taymerlər və sayğaclar yığımıdır. Mikrokontroller, bir qayda olaraq, birdən dördə qədər taymer və sayğacdən ibarətdir. Onlar xarici hadisələrin sayını hesablamaq, müəyyən uzunluqlu (müddətli) siqnallar yaratmaq, kəsilmə sorğuları yaratmaq və s. üçün istifadə edilir. Taymerlər və sayğaclar 8 və 16 bitli olurlar.

## 16-битный 28-pin PDIP PIC24 микроконтроллер



Микроконтроллер Atmel AVR ATmega8 в корпусе DIP



**МІКРОКОНТРОЛЛЕР** – процессордан даһа мукәммәл олmasına бaxmayaraq амма һәләки компьютер дә deyil

**МИКРОКОНТРОЛЛЕР** - это уже не процессор, но ещё и не компьютер.

## MÜNDƏRİCAT

<b>Rəqəmsal elektronika. Giriş.....</b>	<b>2</b>
<b>Məntiqi funksiya haqqında anlayış.....</b>	<b>2</b>
<b>Elementar məntiqi funksiyalar.....</b>	<b>4</b>
<b>Elementar funksiyanın VƏ, VƏ YA, YOX əməliyyatları ilə verilməsi.....</b>	<b>6</b>
<b>Məntiq elementləri haqqında ümumi məlumat.....</b>	<b>9</b>
<b>Triggerlər.Təyinatı, növləri.....</b>	<b>11</b>
<b>Asinxron triggerlər.....</b>	<b>13</b>
<b>Sinxron triggerlər.....</b>	<b>14</b>
<b>Şifratorlar.....</b>	<b>15</b>
<b>Deşifratorlar.....</b>	<b>18</b>
<b>Multipleksor.....</b>	<b>23</b>
<b>Demultipleksor.....</b>	<b>25</b>
<b>Reqistrlər.....</b>	<b>27</b>
<b>Sayğaclar. Cəmləyici ikilik sayğac.....</b>	<b>30</b>
<b>Reversiv sayğaclar.....</b>	<b>32</b>
<b>Cəmləyicilər.....</b>	<b>33</b>
<b>Kod komparatorları.....</b>	<b>36</b>
<b>Mikroprosessorun tipik strukturu.....</b>	<b>39</b>
<b>Mikrokontrollerin tipik struktur sxemi.....</b>	<b>45</b>