

**ПРОГРАМИРУЕМИ ЛОГИЧЕСКИ  
УСТРОЙСТВА.**

**ПРОГРАМИРУЕМИ ПАМЯТИ**

# ПРОГРАМИРУЕМИ ЛОГИЧЕСКИ УСТРОЙСТВА

---

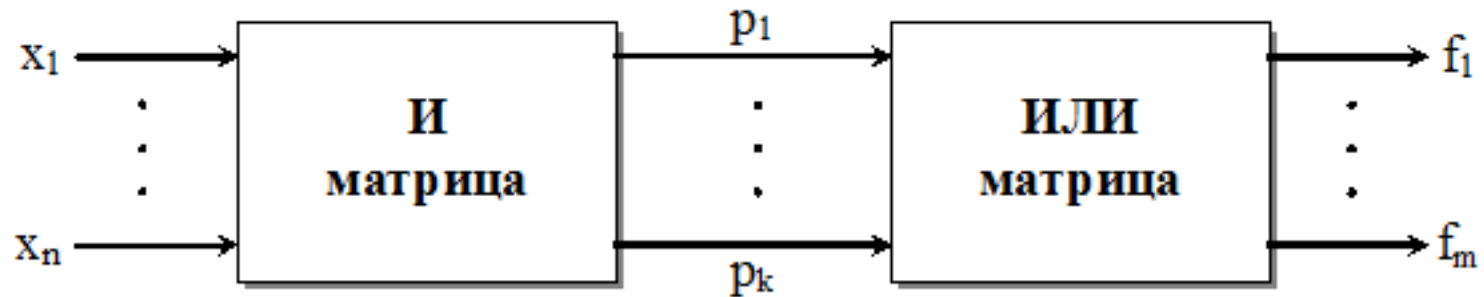
## Въведение

Програмируемите логически устройства имат вътрешна структура, която позволява чрез програмирането им да се реализират комбинационни или последователностни схеми.

# ПРОГРАМИРУЕМИ ЛОГИЧЕСКИ УСТРОЙСТВА

---

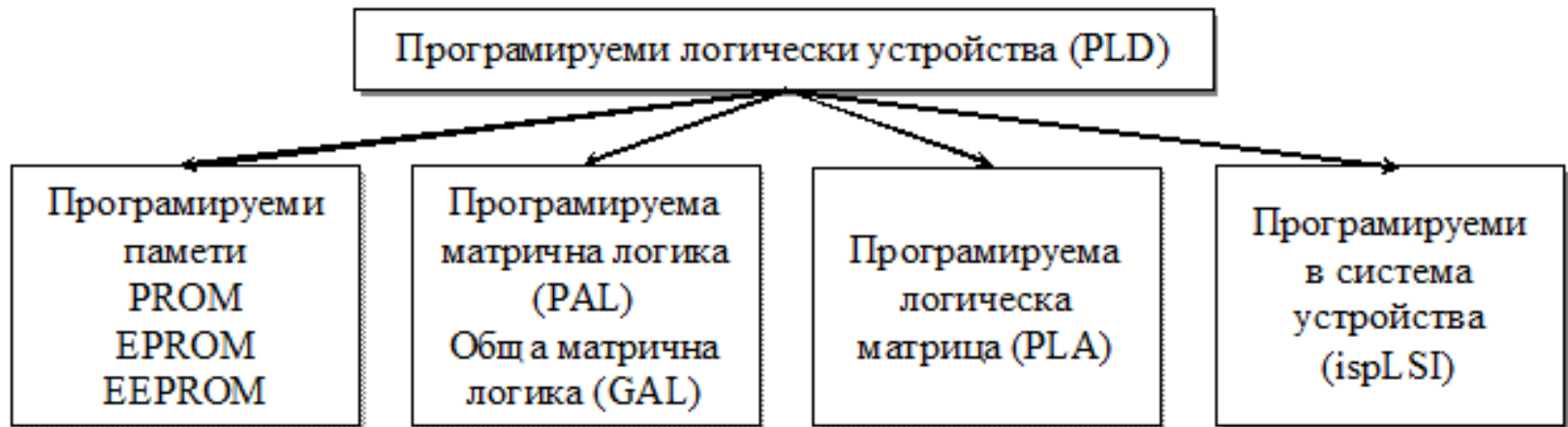
## Структура



# ПРОГРАМИРУЕМИ ЛОГИЧЕСКИ УСТРОЙСТВА

---

Класификация (видове) според това дали е възможно програмирането на “И” и “ИЛИ” матриците.



# ПРОГРАМИРУЕМИ ЛОГИЧЕСКИ УСТРОЙСТВА

---



- *Програмируема памет* - съдържа една непрограмируема “И” матрица, реализираща функция на дешифратор, т.е. генерира пълния набор от минтерми, и една програмируема “ИЛИ” матрица.
- *Програмируема матрична логика (PAL) и обща матрична логика (GAL)* - имат една програмируема “И” матрица и една непрограмируема “ИЛИ” матрица.  
**PAL** позволява еднократно програмиране, а **GAL** позволява многократно програмиране.

# ПРОГРАМИРУЕМИ ЛОГИЧЕСКИ УСТРОЙСТВА



- *Програмируема логическа матрица (PLA)* - съдържа програмируема “И” и програмируема “ИЛИ” матрица.
- *Програмируеми в система устройства (ispLSI)* - имат сложна вътрешна структура и съдържат в себе си голям брой блокове с общо предназначение (наподобяват **PAL, PLA**), блокове за вход/изход, таймери, регистрови блокове и др.

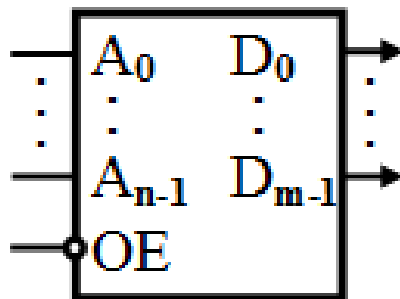
Връзките между отделните блокове, както и връзките в блоковете с общо предназначение се програмират.

За програмирането не се изисква специално апаратно средство (програматор), а може да бъде извършено, докато устройството се намира в състава на общата логическа схема.

# ПРОГРАМИРУЕМИ ПАМЕТИ

---

**ROM (Read Only Memory)** - постоянно запомнящо устройство



Масково програмируеми **ROM** – програмират се еднократно в процеса на производството

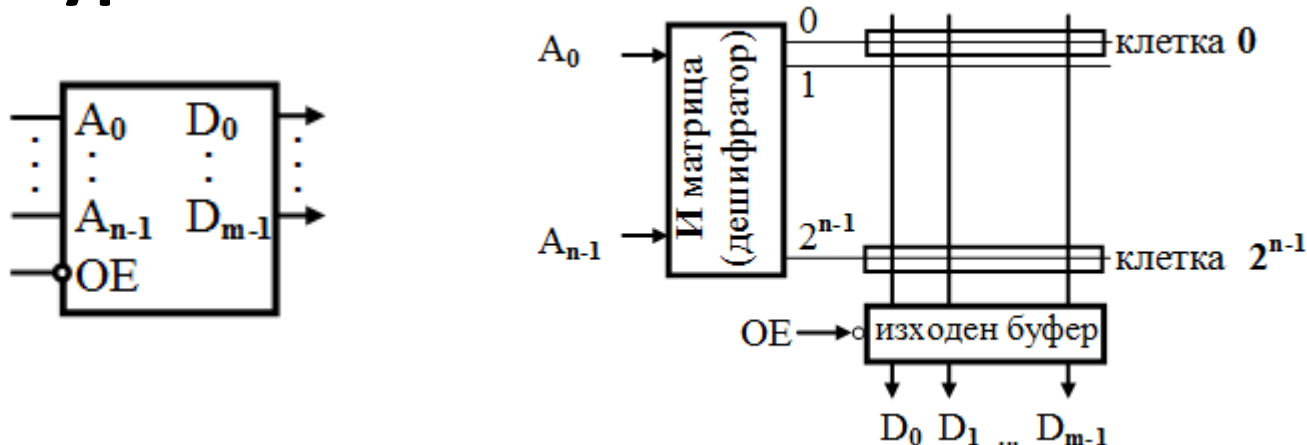
**PROM** - еднократно програмируемо постоянно запомнящо устройство (**ПЗУ**).

**EPROM** - **ПЗУ**, на което може многократно да бъде изтривана информацията с ултравиолетова светлина и да бъде препрограмирано.

**EEPROM** - **ПЗУ**, при което изтриването на информацията е възможно и се извършва по електрически път.

# ПРОГРАМИРУЕМИ ПАМЕТИ

## Структура на ПЗУ



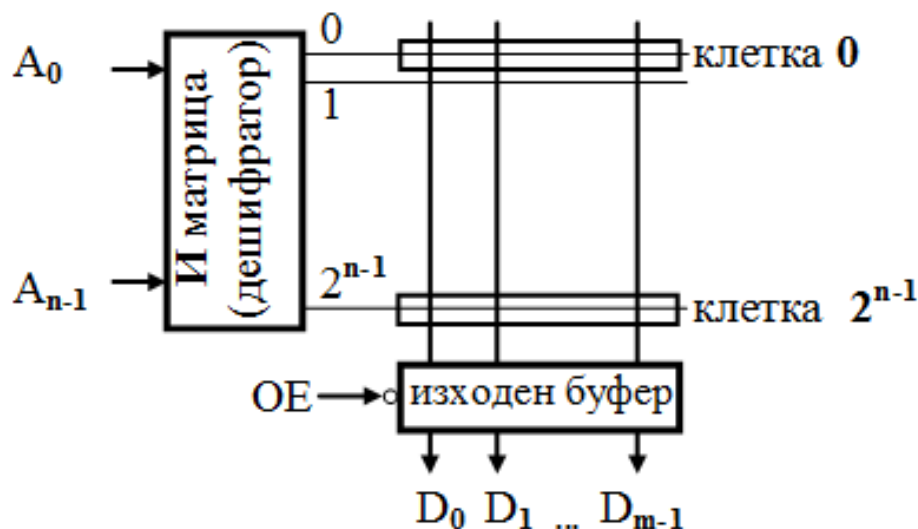
- “И” матрица – непрограмируема, представлява пълен дешифратор.
- “ИЛИ” матрица - програмируема и има  $2^n$  входа,  $m$  изхода и  $2^n$  клетки с по  $m$  възела във всяка от тях.
- Изходен буфер - когато е избран, комутира данновите линии към изходите, а когато сигналът **OE** (**O**utput **E**nable - „разрешаване на изходните буфери”) е пасивен, т. е. “1”, изходите се намират във високо импедансно състояние.



# ПРОГРАМИРУЕМИ ПАМЕТИ

Основни параметри на ПЗУ:

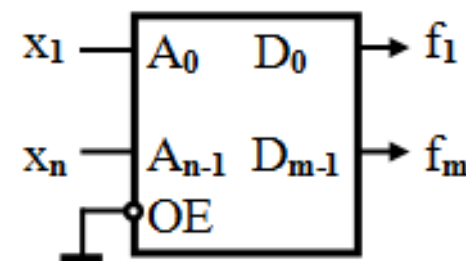
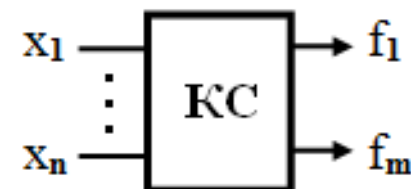
- брой входове ( $n$ ) или брой клетки ( $2^n$ );
- разрядност на клетките ( $m$ ).



# ПРОГРАМИРУЕМИ ПАМЕТИ

## Синтез на комбинационни схеми (КС) на базата на ПЗУ

- Избор на памет с определени параметри и уточняване на информацията, която трябва да се запише в нея, за да се генерират необходимите изходни функции.
- Броят на **адресните входове** на избраното **ПЗУ** трябва да бъде **по-голям или равен** на броя на **входните променливи** на комбинационната схема.
- Броят на **данните изходи** на **ПЗУ** трябва да бъде **по-голям или равен** на броя на **изходните функции** на комбинационната схема.



# ПРОГРАМИРУЕМИ ПАМЕТИ

---

**Синтез на комбинационни схеми (КС) на базата на ПЗУ**

Програмирането на **ПЗУ** става чрез специализирани устройства - програматори.

Необходимо е само да се определи какви данни ще се записват в клетките на паметта.

# ПРОГРАМИРУЕМИ ПАМЕТИ

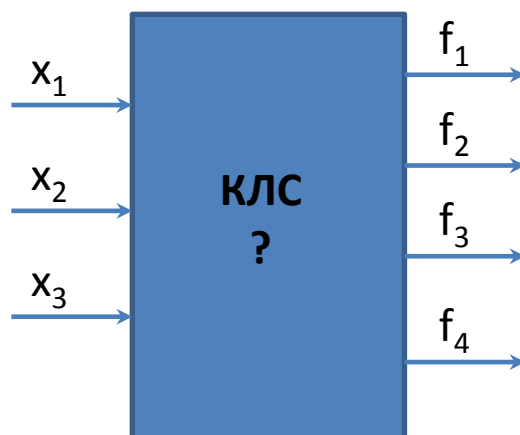
**Пример:** На базата на **ПЗУ** да се синтезира **КС**, реализираща следните функции:

$$f_1(x_1, x_2, x_3) = Vm(1, 3, 5)$$

$$f_2(x_1, x_2, x_3) = Vm(4, 5, 6, 7)$$

$$f_3(x_1, x_2, x_3) = Vm(2, 3, 6, 7)$$

$$f_4(x_1, x_2, x_3) = Vm(0, 3, 4)$$

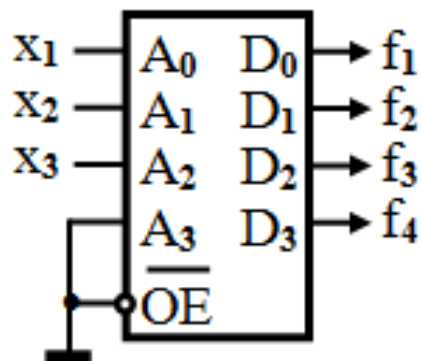


# ПРОГРАМИРУЕМИ ПАМЕТИ

## Решение:

1. Избираме **ПЗУ** за което  $n \geq 3$  ( $x_1, x_2, x_3$ ), т. е. то трябва да има минимум три адресни входа (**8** клетки) и  $m \geq 4$  ( $f_1, f_2, f_3, f_4$ ), т. е. изходите трябва да бъдат минимум четири.

От каталог избираме **PROM** с организация **16x4**, т. е. **16** клетки по четири бита във всяка клетка.



# ПРОГРАМИРУЕМИ ПАМЕТИ

---

## Решение:

2. Присвояваме входните променливи и изходните функции съответно на адресните входове и данновите изходи.
3. Съставяме таблицата на истинност на системата логически функции и определяме данните, които трябва да се запишат в съответните клетки на **ПЗУ**.

# ПРОГРАМИРУЕМИ ПАМЕТИ

## Решение:

$$f_1(x_1, x_2, x_3) = Vm(1, 2, 5)$$

$$f_2(x_1, x_2, x_3) = Vm(4, 5, 6, 7)$$

$$f_3(x_1, x_2, x_3) = Vm(2, 3, 6, 7)$$

$$f_4(x_1, x_2, x_3) = Vm(0, 3, 4)$$

	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	адрес	данные
	x <sub>3</sub>	x <sub>2</sub>	x <sub>1</sub>	f <sub>4</sub>	f <sub>3</sub>	f <sub>2</sub>	f <sub>1</sub>		
0	0	0	0	1	0	0	0	0	8
1	0	0	1	0	0	0	1	1	1
2	0	1	0	0	1	0	1	2	5
3	0	1	1	1	1	0	0	3	C
4	1	0	0	1	0	1	0	4	A
5	1	0	1	0	0	1	1	5	3
6	1	1	0	0	1	1	0	6	6
7	1	1	1	0	1	1	0	7	6

# ПРОГРАМИРУЕМИ ПАМЕТИ

## Решение:

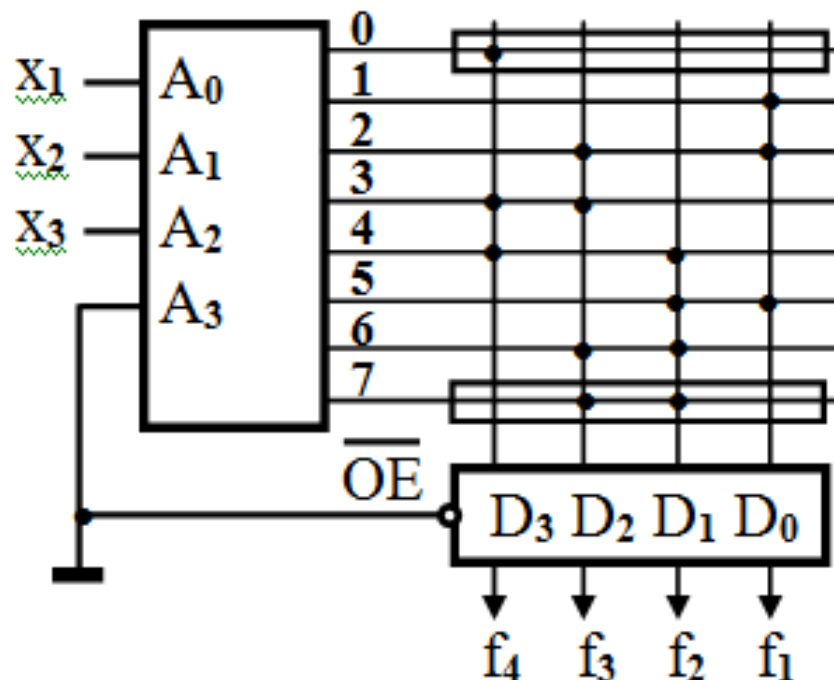
$$f_1(x_1, x_2, x_3) = \vee m(1, 2, 5)$$

$$f_2(x_1, x_2, x_3) = \vee m(4, 5, 6, 7)$$

$$f_3(x_1, x_2, x_3) = \vee m(2, 3, 6, 7)$$

$$f_4(x_1, x_2, x_3) = \vee m(0, 3, 4)$$

	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	адрес	данные
	x <sub>3</sub>	x <sub>2</sub>	x <sub>1</sub>	f <sub>4</sub>	f <sub>3</sub>	f <sub>2</sub>	f <sub>1</sub>		
0	0	0	0	1	0	0	0	0	8
1	0	0	1	0	0	0	1	1	1
2	0	1	0	0	1	0	1	2	5
3	0	1	1	1	1	0	0	3	C
4	1	0	0	1	0	1	0	4	A
5	1	0	1	0	0	1	1	5	3
6	1	1	0	0	1	1	0	6	6
7	1	1	1	0	1	1	0	7	6





# ПРОГРАМИРУЕМИ ПАМЕТИ

---

## Обобщение:

При синтез на комбинационни схеми с **ПЗУ** *минимизация на логическите функции не се извършва.*

Тук липсва многостъпалност на логическата схема и съответно състезания.

Времето за реакция на комбинационната схема е **еднакво** за всяка отделна функция и зависи от бързодействието на избраното **ПЗУ**.

Обикновено това време е около  $10 \div 200$  ns.

# ПРОГРАМИРУЕМИ ПАМЕТИ

---

**Синтез на последователности схеми**  
(логически схеми с памет) **на базата на ПЗУ**

Синтезът на логическа схема с памет се свежда до избор на тригери за паметта и до реализация само на комбинационната схема чрез **ПЗУ**.

# ПРОГРАМИРУЕМИ ПАМЕТИ

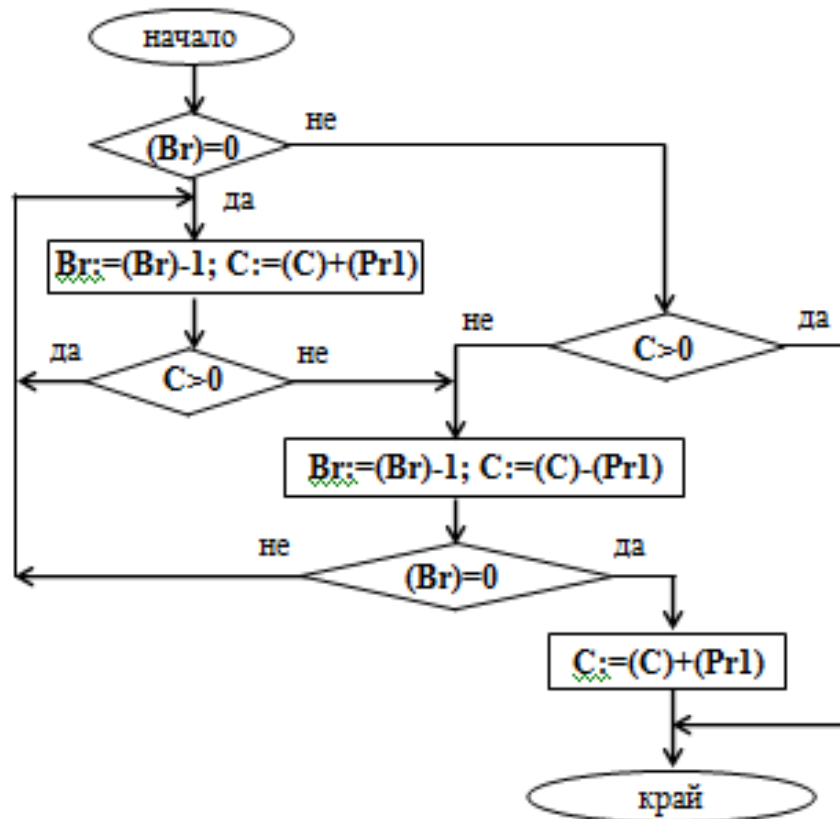
---

**Синтез на последователности схеми**  
(логически схеми с памет) **на базата на ПЗУ**

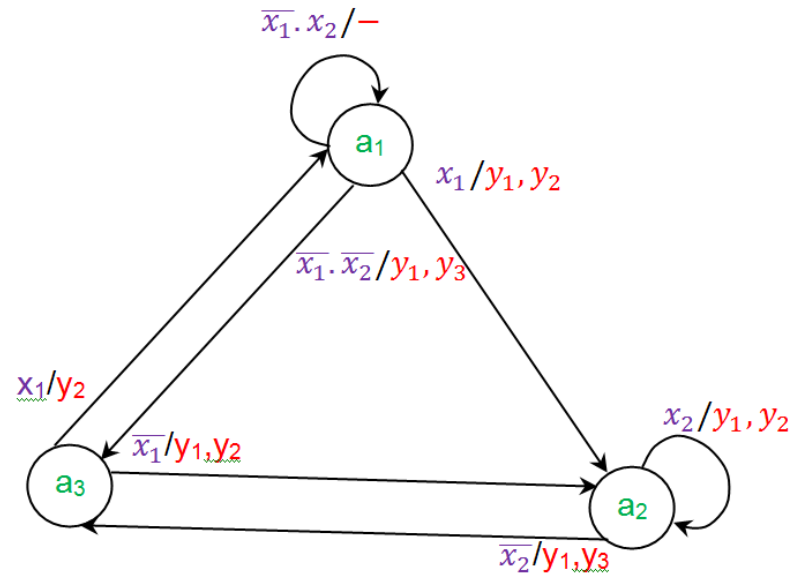
Синтезът на логическа схема с памет се свежда до избор на тригери за паметта и до реализация само на комбинационната схема чрез **ПЗУ**.

# ПРОГРАМИРУЕМИ ПАМЕТИ

**Пример:** Да се синтезира микропрограмен автомат на Мили, зададен с БСА.

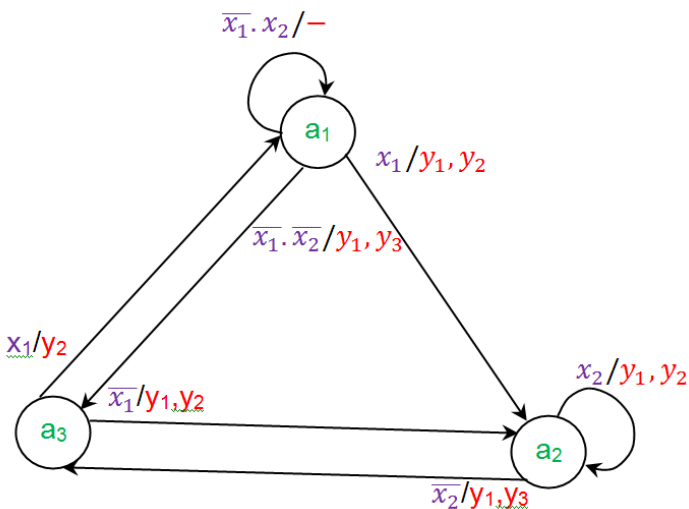


## Решение



# ПРОГРАМИРУЕМИ ПАМЕТИ

## Решение



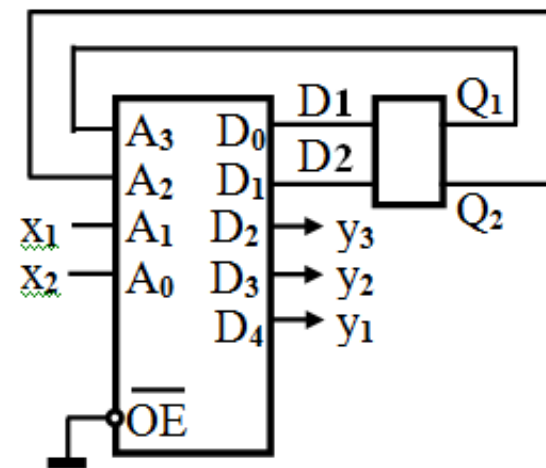
текущо състояние			изходи	входове	ново състояние			функции	
$a(t)$	$Q_1$	$Q_2$	$y(t)$	$x(t)$	$a(t+1)$	$Q_1$	$Q_2$	$D_1$	$D_2$
$a_1$	0	0	$y_1, y_2$	$x_1$	$a_2$	0	1	0	1
			-	$\overline{x_1} \cdot x_2$	$a_1$	0	0	0	0
			$y_1, y_3$	$\overline{x_1} \cdot \overline{x_2}$	$a_3$	1	0	1	0
$a_2$	0	1	$y_1, y_3$	$\overline{x_2}$	$a_3$	1	0	1	0
			$y_1, y_2$	$x_2$	$a_2$	0	1	0	1
$a_3$	1	0	$y_1, y_2$	$\overline{x_1}$	$a_2$	0	1	0	1
			$y_2$	$x_1$	$a_1$	0	0	0	0

## Решение

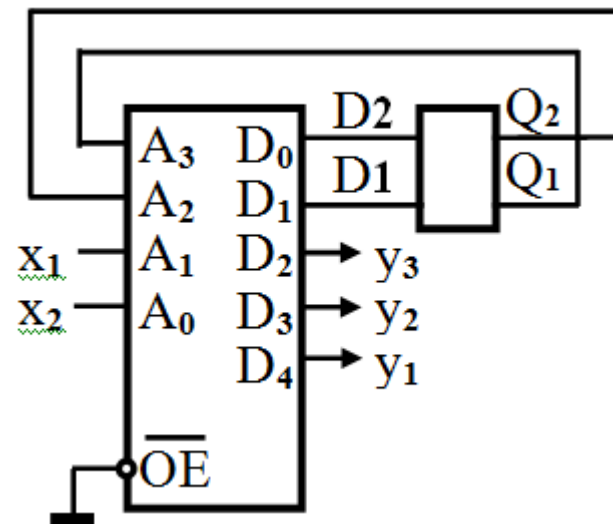
От нея се уточнява броят на входовете и изходите на **ПЗУ**.

В случая имаме четири входа и пет изхода. Избираме **ПЗУ** с организация **16x8**.

текущо състояние			изходи	входове	ново състояние			функции	
$a(t)$	$Q_1$	$Q_2$	$y(t)$	$x(t)$	$a(t+1)$	$Q_1$	$Q_2$	$D_1$	$D_2$
$a_1$	0	0	$y_1, y_2$	$x_1$	$a_2$	0	1	0	1
			-	$\overline{x_1}, x_2$	$a_1$	0	0	0	0
			$y_1, y_3$	$\overline{x_1}, \overline{x_2}$	$a_3$	1	0	1	0
$a_2$	0	1	$y_1, y_3$	$\overline{x_2}$	$a_3$	1	0	1	0
			$y_1, y_2$	$x_2$	$a_2$	0	1	0	1
$a_3$	1	0	$y_1, y_2$	$\overline{x_1}$	$a_2$	0	1	0	1
			$y_2$	$x_1$	$a_1$	0	0	0	0



текущо състояние			изходи	входове	ново състояние			функции	
$a(t)$	$Q_1$	$Q_2$	$y(t)$	$x(t)$	$a(t+1)$	$Q_1$	$Q_2$	$D_1$	$D_2$
$a_1$	0	0	$y_1, y_2$	$x_1$	$a_2$	0	1	0	1
			-	$\overline{x_1}, x_2$	$a_1$	0	0	0	0
			$y_1, y_3$	$\overline{x_1}, \overline{x_2}$	$a_3$	1	0	1	0
$a_2$	0	1	$y_1, y_3$	$\overline{x_2}$	$a_3$	1	0	1	0
			$y_1, y_2$	$x_2$	$a_2$	0	1	0	1
$a_3$	1	0	$y_1, y_2$	$\overline{x_1}$	$a_2$	0	1	0	1
			$y_2$	$x_1$	$a_1$	0	0	0	0

[illegible]