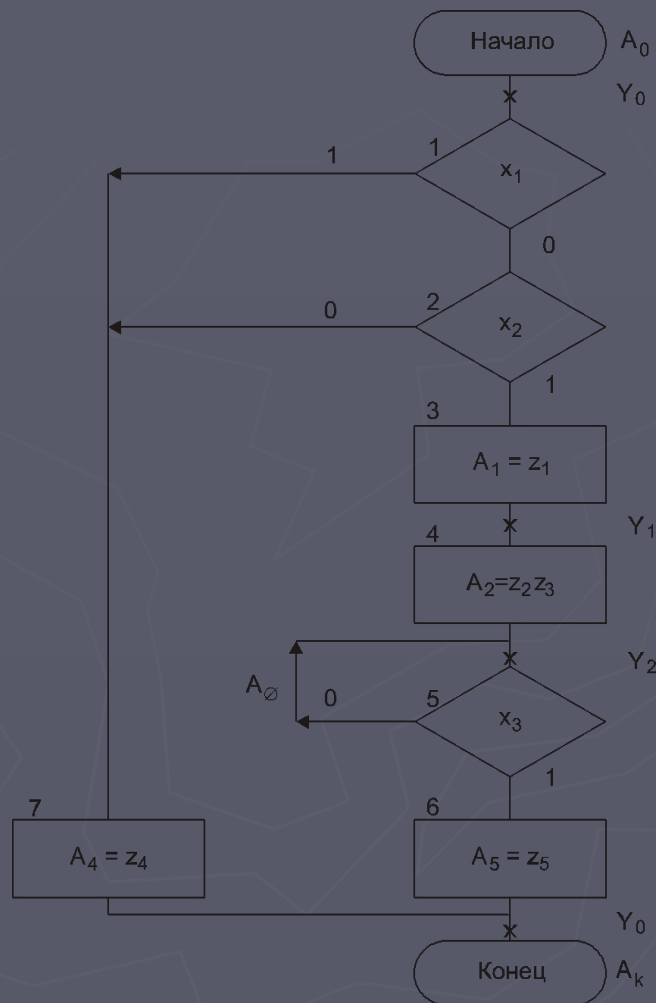


Лекция 14
Синтез МПУУ на «гибкой» логике

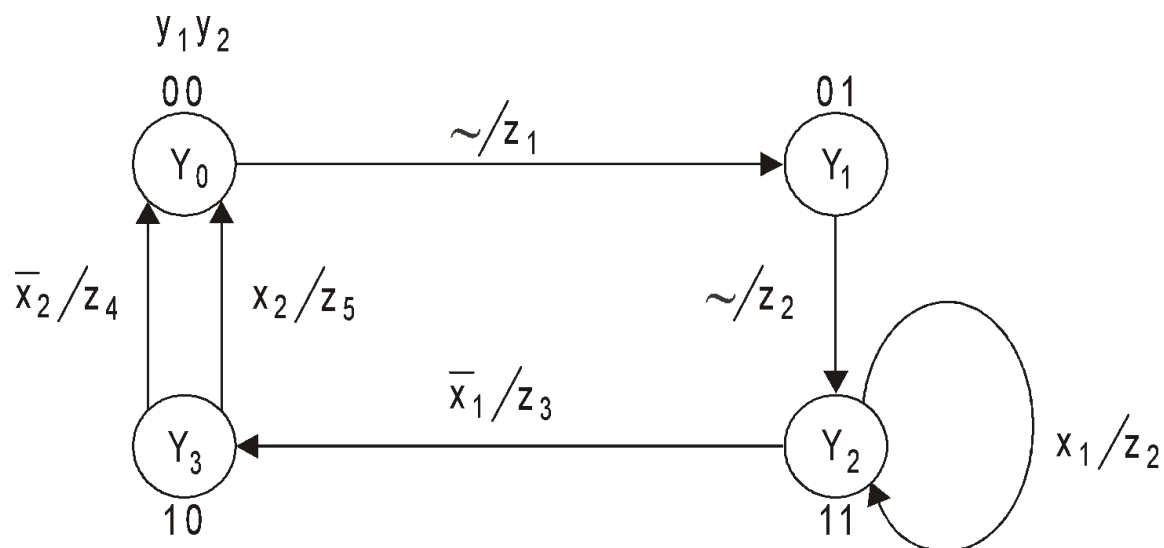
- **1.Синтез МПУУ на ПЗУ**
- **2.Синтез МПУУ на ПЗУ и мультиплексоре**
- **3.Синтез МПУУ со счётчиком микрокоманд**

1. Синтез МПУУ на ПЗУ

► Дана ГСА



Построим граф автомата



Переход от графа автомата к таблице программирования ПЗУ

- Схема может быть построена на основе постоянного запоминающего устройства (ПЗУ).
- В этом случае нет необходимости в минимизации переключательных функций, так как они представляются в СДНФ. Поэтому после получения ОГСА, сразу строится таблица программирования ПЗУ: сначала обобщенная (с «тильдами»), затем полная.

Построим обобщенную таблицу переходов-выходов.

Используем d-триггеры – со входами d2, d1.

- Обобщенная таблица переходов-выходов

y2	y1	x2	x1	y2(t+1) d2(t)	y1(t+1) d1(t)	Микрооперации				
						z1	z2	z3	z4	z5
0	0	~	~	0	1	1	0	0	0	0
0	1	~	~	1	1	0	1	0	0	0
1	1	~	0	1	0	0	0	1	0	0
1	1	~	1	1	1	0	1	0	0	0
1	0	0	~	0	0	0	0	0	1	0
1	0	1	~	0	0	0	0	0	0	1

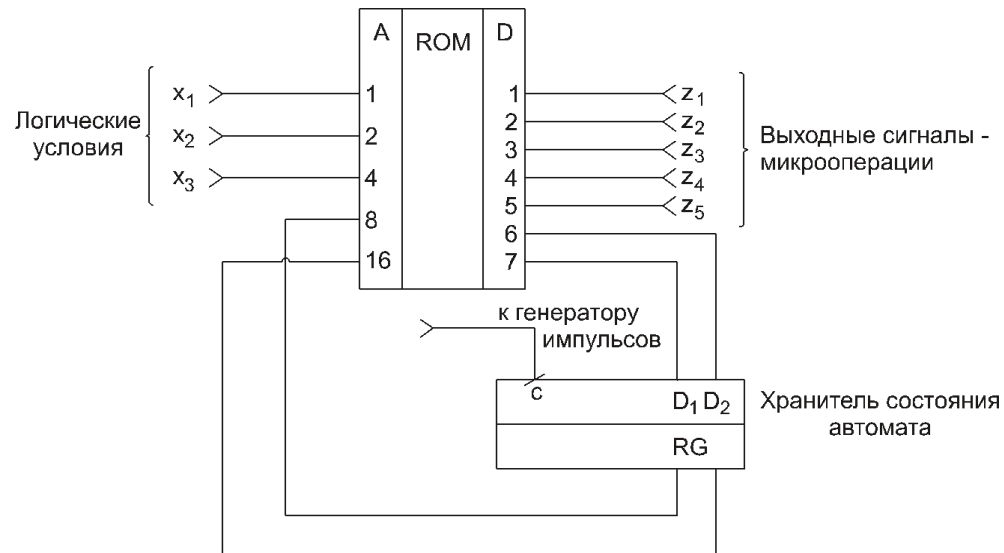
Таблица программирования ПЗУ, построенная по обобщенной таблице переходов-выходов

- ТППЗУ

Адреса				Данные						
y2	y1	x2	x1	d2(t)	d1(t)	Микрооперации				
						z1	z2	z3	z4	z5
0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	0	1	1	0	0	0	0
0	0	1	1	0	1	1	0	0	0	0
0	1	0	0	1	1	0	1	0	0	0
0	1	0	1	1	1	0	1	0	0	0
0	1	1	0	1	1	0	1	0	0	0
0	1	1	1	1	1	0	1	0	0	0
1	1	0	0	1	0	0	0	1	0	0
1	1	1	0	1	0	0	0	1	0	0
1	1	0	1	1	1	0	1	0	0	0
1	1	1	1	1	1	0	1	0	0	0
1	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	0	1
1	0	1	1	0	0	0	0	0	0	1

Схема на ПЗУ

- МПУУ ПЗУ



ПЗУ можно моделировать на базе дешифратора 74154.

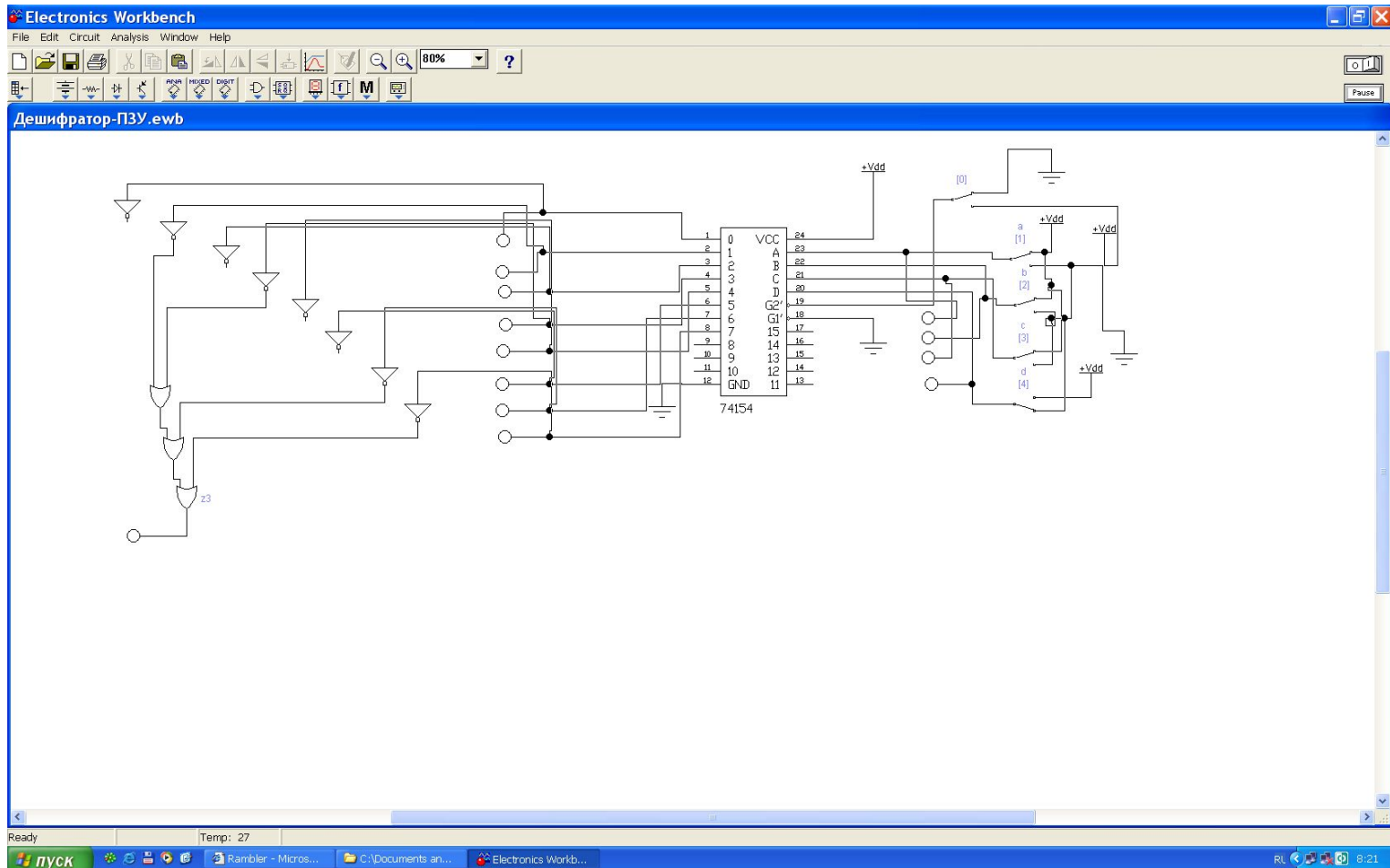


Схема на ПЗУ

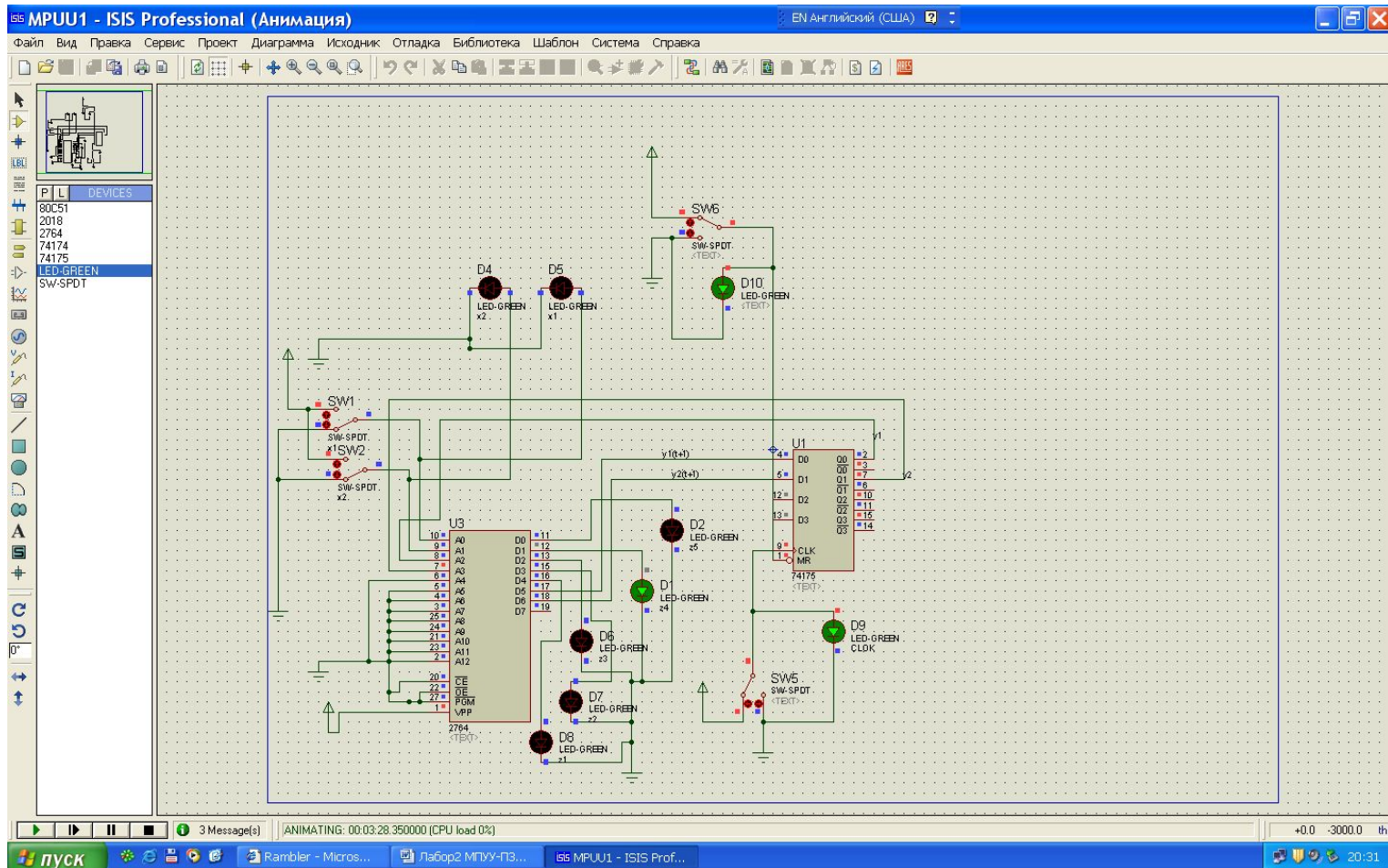


Схема на ПЗУ

- Такая же схема может быть построена для автомата-распознавателя, счётчика и т.д.

2. МПУУ на ПЗУ и мультиплексоре

- Недостатком МПУУ на базе ПЗУ является большой объем памяти, необходимой при большом количестве логических условий. Сократить объем памяти ПЗУ можно путем использования так называемых мультиплексоров, или коммутаторов каналов, позволяющих в каждом состоянии проверять только одно₁ или несколько (не все сразу)

2. МПУУ на ПЗУ и мультиплексоре

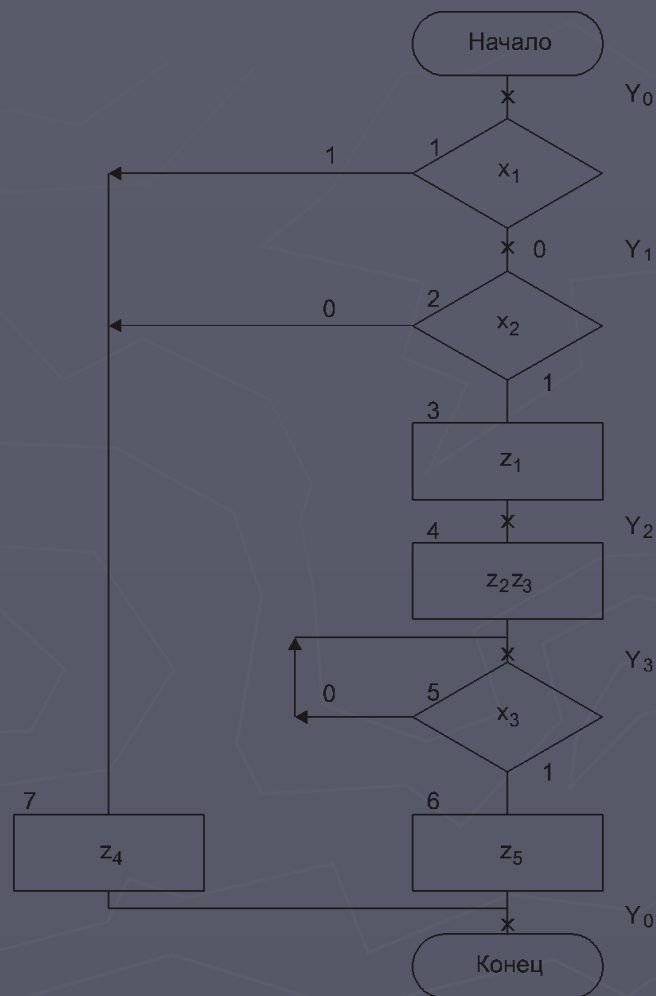
- В этом случае будут особенности отметки ГСА (при получении ОГСА). Если за условной вершиной следует условие, с целью проверки только одного условия метка ставится после данной условной вершины. В остальном разметка ГСА не отличается от разметки для автомата без мультиплексора. Эти особенности обуславливаются требованиями проверки в каждом такте (на каждом переходе) не более одного логического условия.

2. МПУУ на ПЗУ и мультиплексоре

- Мультиплексор адресуется элементами памяти, поэтому на его выходе присутствуют некоторые переменные или заранее оговоренная константа. Сигнал на выходе мультиплексора назовем псевдопеременной (х), причем договоримся сопоставлять с безусловным переходом константу нуля (на соответствующий вход мультиплексора подаем константу нуля).

Разметка ГСА для схемы с мультиплексором

Видно, что переход
от метки к метке
осуществляется
с проверкой не
более одного
логического условия



Граф автомата для схемы с MS (MUX)

Теперь кодируем вершины графа. Число элементов памяти не изменилось (y2y1).

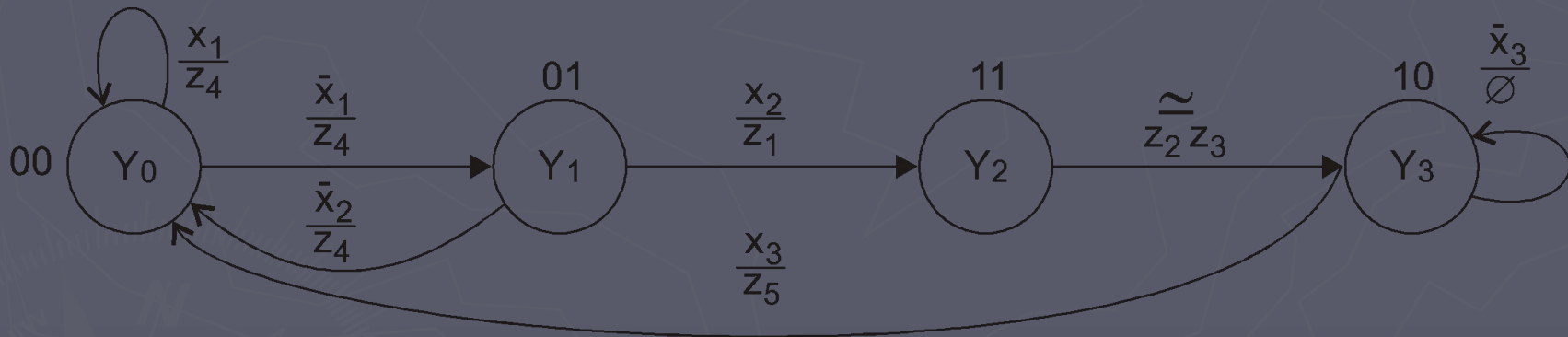


Таблица программирования

- Построим таблицу программирования ПЗУ. Сразу строится полная таблица программирования. Мультиплексор адресуется элементами памяти, поэтому на его выходе присутствуют некоторые переменные или заранее оговоренная константа. Сигнал на выходе мультиплексора назовем псевдопеременной (x), причем договоримся сопоставлять с безусловным переходом константу нуля (на соответствующий вход мультиплексора подаем константу нуля).
- Количество строк таблицы программирования ПЗУ равно количеству дуг графа переходов, количество адресных разрядов для одновыходного мультиплексора равно $s+1$, где s — число элементов памяти. Количество разрядов микроопераций не изменяется

Таблица программирования

► ТППЗУ

Адрес			Данные								
	y_2	y_1	x	z_1	z_2	z_3	z_4	z_5	$y_2(t+1)$	$y_1(t+1)$	
x_1	0	0	0	0	0	0	0	0	0	1	$Y_0 \rightarrow Y_1$
	0	0	1	0	0	0	1	0	0	0	$Y_0 \rightarrow Y_0$
x_2	0	1	0	0	0	0	1	0	0	0	$Y_1 \rightarrow Y_0$
	0	1	1	1	0	0	0	0	1	1	$Y_1 \rightarrow Y_2$
$\sim$	1	1	0	0	1	1	0	0	1	0	$Y_2 \rightarrow Y_3$
x_3	1	0	1	0	0	0	0	1	0	0	$Y_3 \rightarrow Y_0$
	1	0	0	0	0	0	0	0	1	0	$Y_3 \rightarrow Y_3$

Полная таблица программирования ПЗУ для автомата с мультиплексором в 16-ричном коде:

Адрес			Данные								
	y_2	y_1	x	z_1	z_2	z_3	z_4	z_5	$y_2(t+1)$	$y_1(t+1)$	
x_1	0	0	0	0	0	0	0	0	0	1	$Y_0 \rightarrow Y_1$
	0	0	1	0	0	0	1	0	0	0	$Y_0 \rightarrow Y_0$
x_2	0	1	0	0	0	0	1	0	0	0	$Y_1 \rightarrow Y_0$
	0	1	1	1	0	0	0	0	1	1	$Y_1 \rightarrow Y_2$
$\sim$	1	1	0	0	1	1	0	0	1	0	$Y_2 \rightarrow Y_3$
x_3	1	0	1	0	0	0	0	1	0	0	$Y_3 \rightarrow Y_0$
	1	0	0	0	0	0	0	0	1	0	$Y_3 \rightarrow Y_3$

Адрес:	Данные
0_{16}	01_{16}
1_{16}	08_{16}
2_{16}	08_{16}
3_{16}	43_{16}
6_{16}	32_{16}
5_{16}	04_{16}
4_{16}	02_{16}

Получим таблицу подключения информационных входов мультиплексора.

- Ее можно составить по графу автомата. В состоянии $y_0(00)$ проверяется логическое условие x_1 , в состоянии $y_1(01)$ проверяется логическое условие x_2 , в состоянии $y_2(11)$ не проверяются логические условия – к соответствующему входу мультиплексора подключим генератор «0», в состоянии $y_3(10)$ проверяется логическое условие x_3 .

Таблица распределения входов мультиплексора

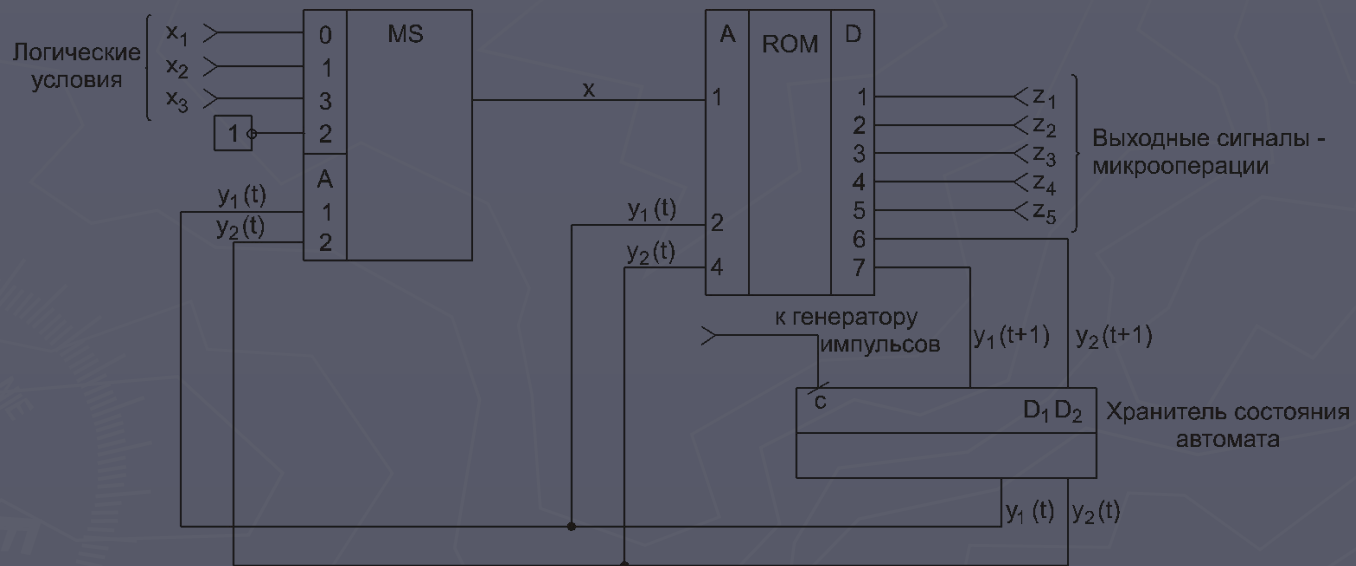
- ТРВ MS

Номер информа ционного входа	Переменная (константа)
0_{10}	x_1
1_{10}	x_2
3_{10}	«0»
2_{10}	x_3

~

Схема МПУУ на ПЗУ и MS с тремя входными переменными, двумя переменными состояния и пятью микрооперациями

► МПУУ на ПЗУ и MS



Здесь входы данных обозначены E , адресные входы – A , B , C , D . Выход данных – W , разрешение – G' .

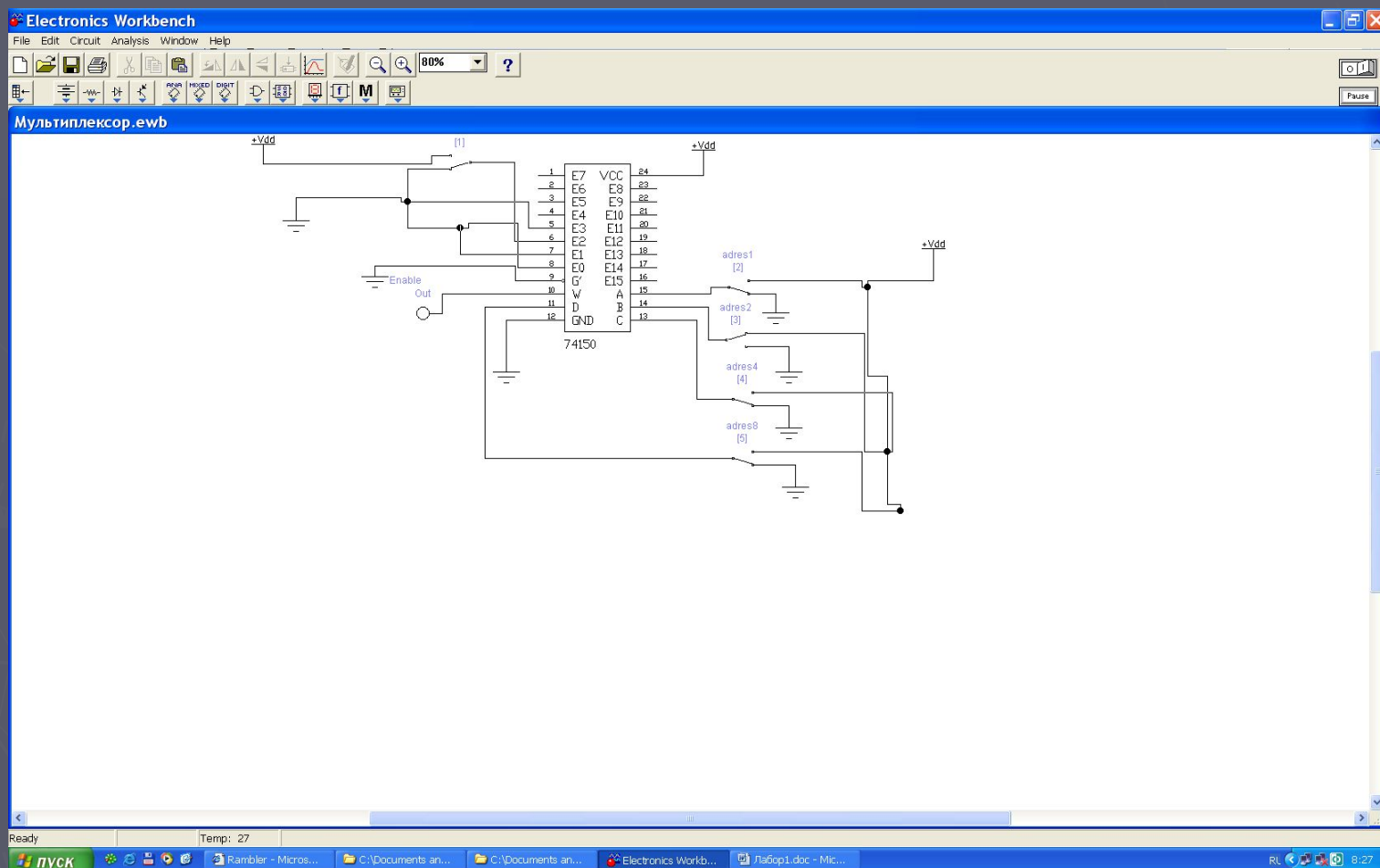
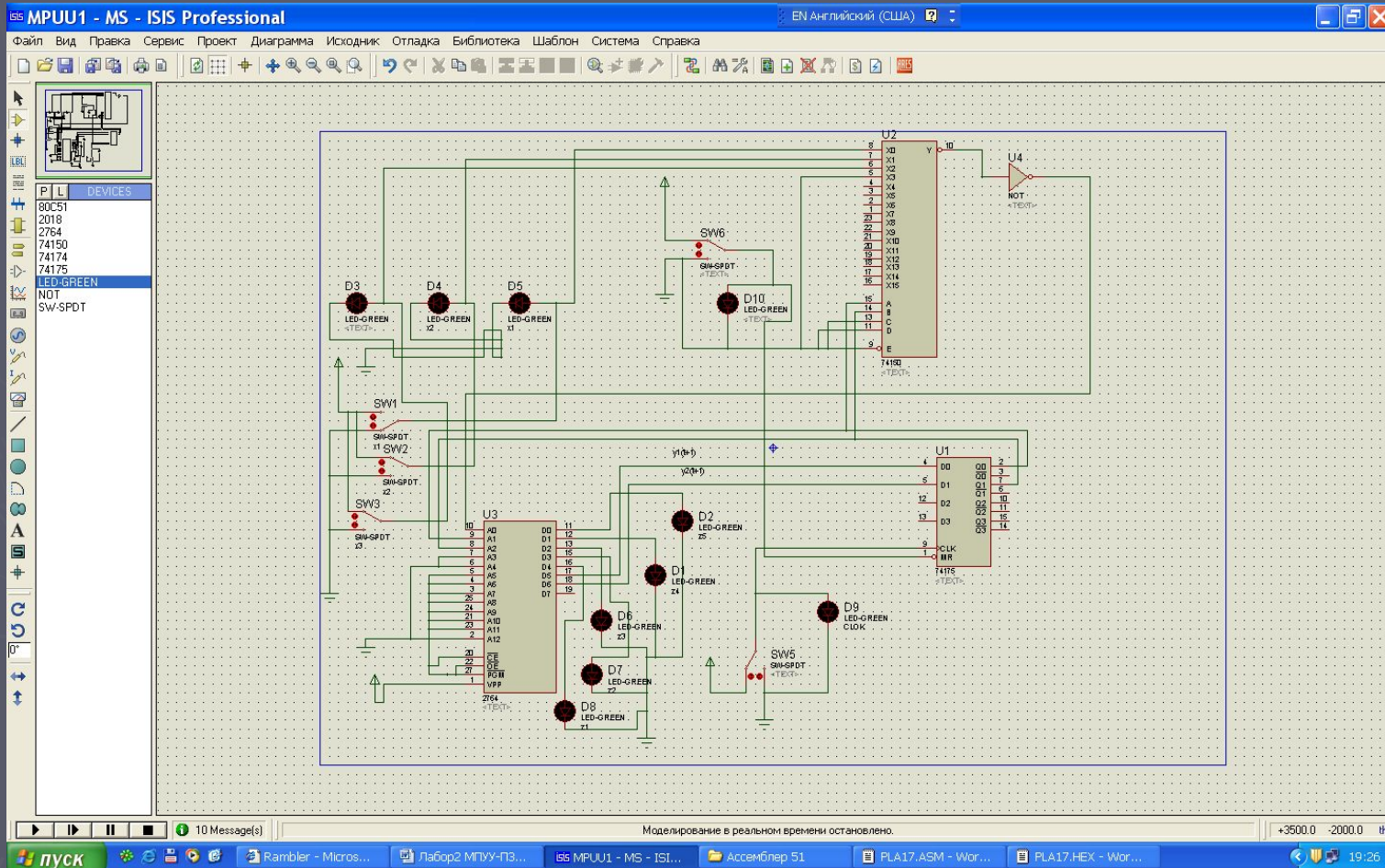


Схема МПУУ на ПЗУ и МС



Оценка реализации

- Объем памяти у такого автомата сокращается – необходимо $7 \cdot 7 = 49$ бит (против 168 бит для автомата на основе ПЗУ).
- Здесь мультиплексор в зависимости от состояния автомата, которое адресует не только ПЗУ, но и подключаемый к мультиплексору канал, передает на вход ПЗУ одну из переменных, или константу, в случае безусловного перехода.

3. Синтез МПУУ со счётчиком микрокоманд

Разделение микрокоманд на два типа

Дальнейшее углубление в область программного обеспечения заключается в сопоставлении каждому блоку ГСА некоторой микрокоманды

в простейшем случае формируется из микрокоманд двух типов :

1) операционной

Признак микрокоманды $ПМ(p)=1$	Микрооперации
-----------------------------------	---------------

2) специальной или перехода

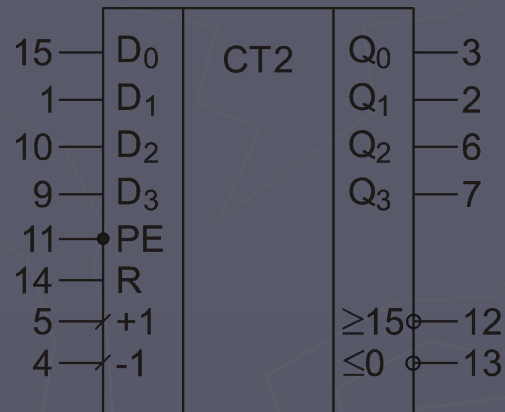
Признак микрокоманды $ПМ(p)=0$	Код логического условия	Адрес перехода
-----------------------------------	-------------------------	----------------

Счётчик микрокоманд

- Если выполняется операционная микрокоманда, то последовательно выбираются адреса постоянного запоминающего устройства (ПЗУ) и последовательно выдаются микрооперации в операционное устройство. Это может быть реализовано путем перебора состояний счетчика микрокоманд по счетному входу +1

Счётчик с предустановкой

► СТ2

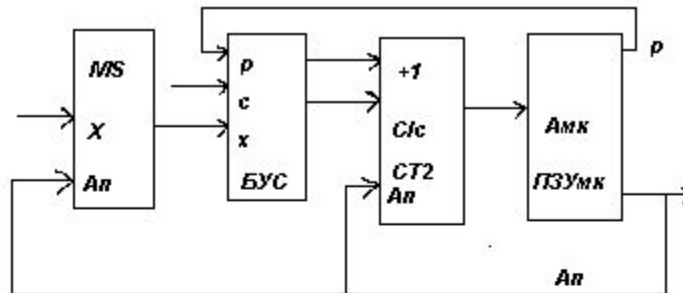


Блок микропрограммного управления со счётчиком микрокоманд

- В случае выполнения специальной микрокоманды $p = 0$, если логическое условие x равно единице, – производится переход по указанному адресу с использованием входа предустановки PE (clс) по указанным в поле адреса данным D.
- Если логическое условие x равно нулю, то выбирается следующий адрес. Если номер логического условия равен нулю, то это безусловный переход, и производится передача управления по указанному адресу (на входах D) (как будто $x=1$) .
- Такая логика управления реализуется блоком микропрограммного управления со счетчиком микрокоманд, который адресует память микропрограмм.

Простейший БМУ

- БМУ



***P** - признак микрокоманды*

***ПЗУмк** – память микрокоманд **СТ2** – счётчик микрокоманд*

***Амк** - адрес очередной микрокоманды*

Ап** – адрес перехода **БУС** – блок управления **СТ2

***x** – текущая переменная **MS** – мультиплексор*

***X** - переменные*

***c** – синхросигнал тактового генератора*

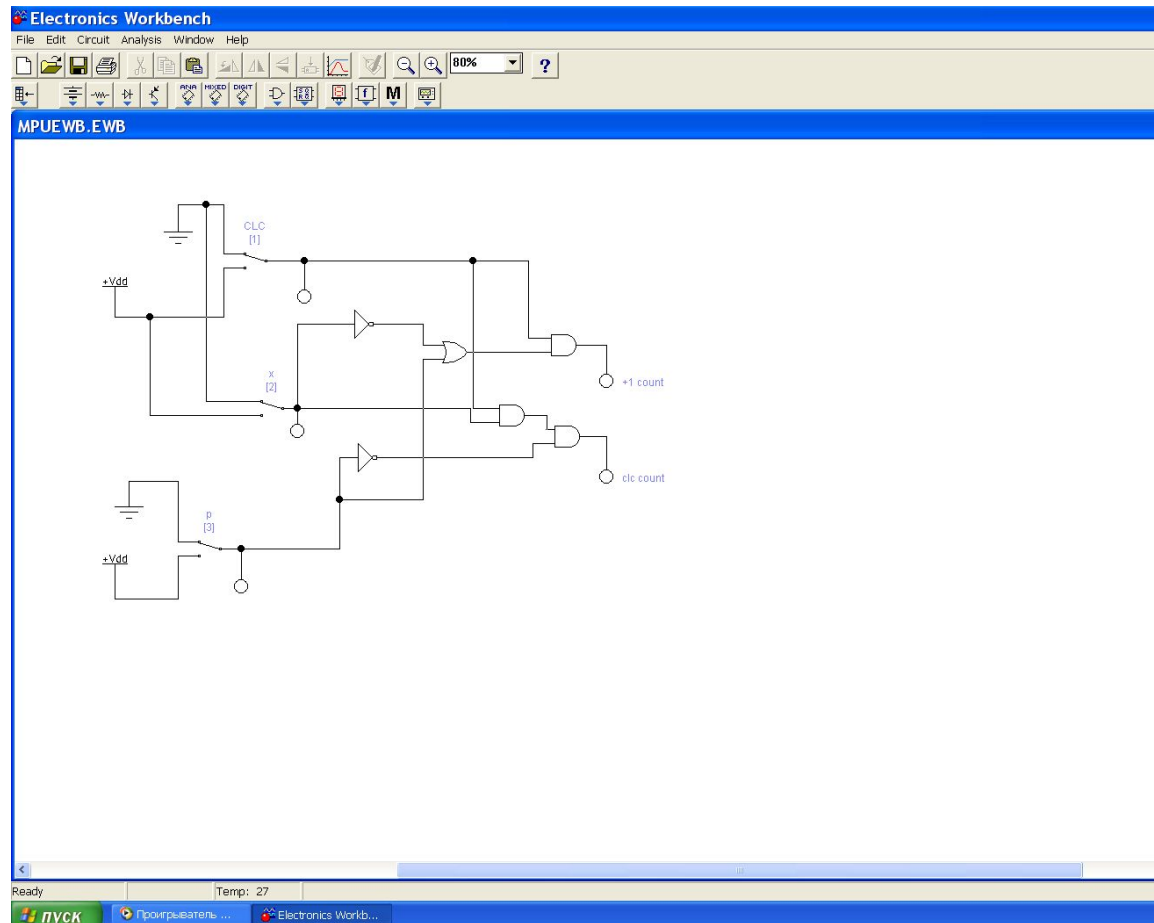
***+1** – счёт, выбор очередного адреса, инкремент*

***C/c** – переход по заданному адресу*

Блок управления счётчиком для БМУ

- БУС

$$\begin{aligned}f(+1) &= \overline{p}\overline{c}\overline{x} \vee p\overline{c} = \\ &= c(\overline{p}\overline{x} \vee p) = c(\overline{x} \vee p) \\ f(clc) &= \overline{p}c\overline{x}\end{aligned}$$



Использование типовых БМУ в МПК

- В микропроцессорных комплектах многокристальных (секционированных, микропрограммируемых, масштабируемых), где можно получить любую разрядность обрабатываемых данных, блоки микропрограммного управления выполняются в виде отдельной БИС
- Кроме того необходим регистр признаков процессорного блока, мультиплексор признаков, запоминающее устройство микрокоманд, регистр микрокоманд и специальный преобразователь начального адреса для получения начального адреса микропрограммы по коду операции (команды)
- Микропрограммирование – весьма трудоёмкий процесс

ГСА

► Дана ГСА

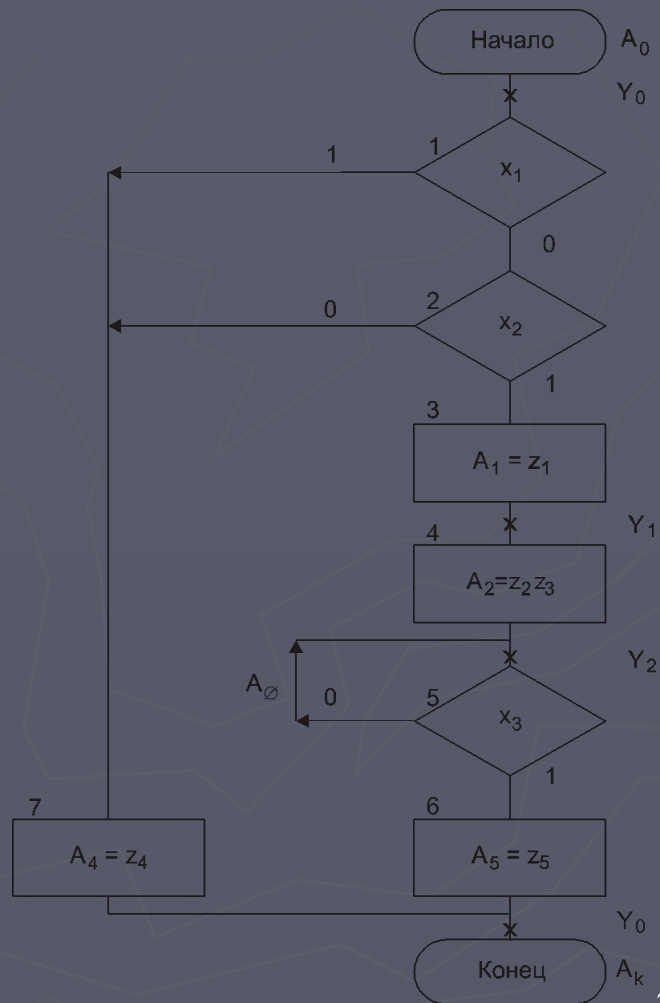
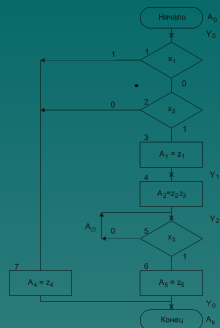
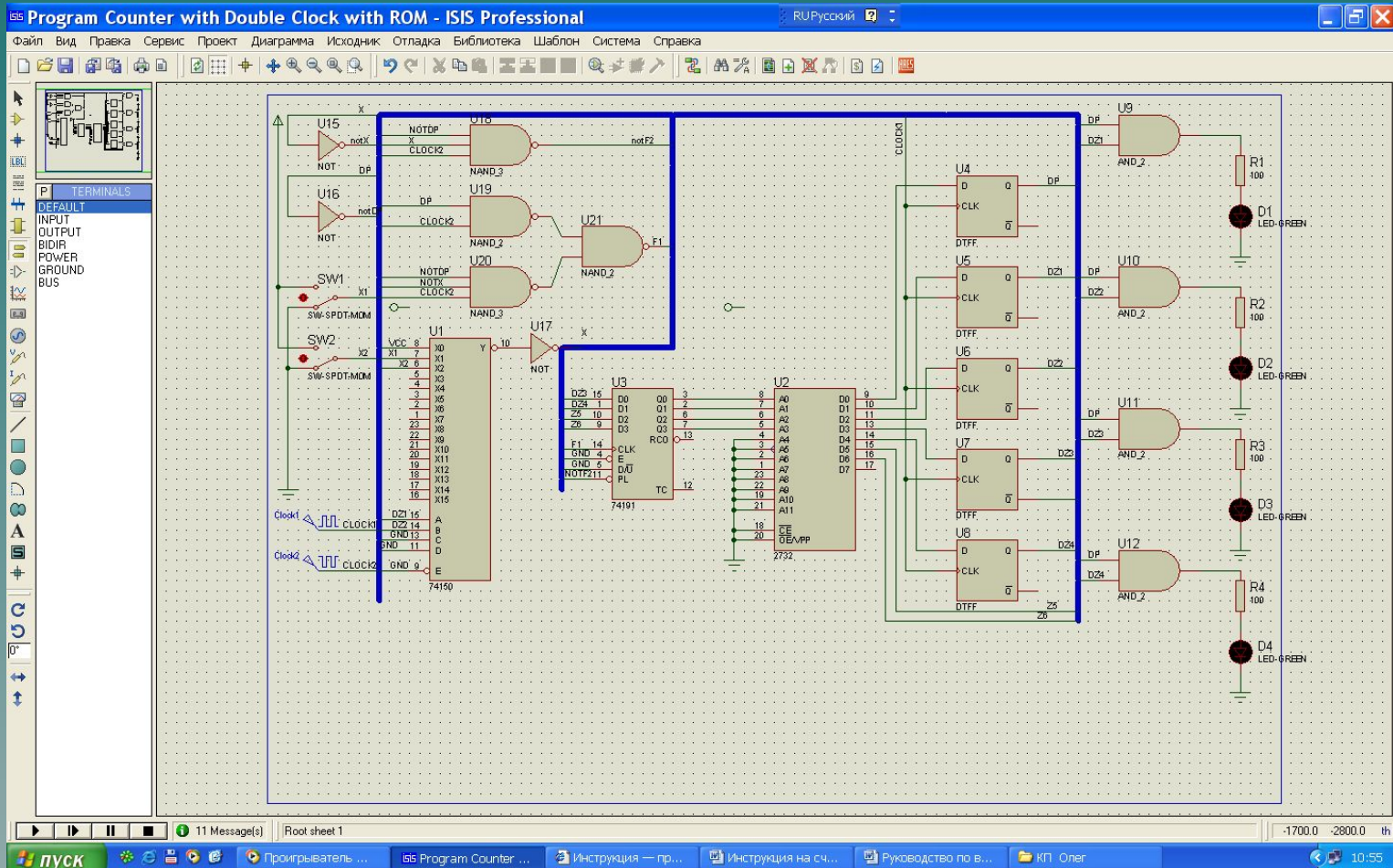


Таблица программирования ПЗУ



Мет ка	Адрес Микроко манда				Микрокоманда								Комментарий
					П М	1	2	3	4	5	6	7	
M0:	0	0	0	0	0	0	1	0	1	1	0	0	Переход, если $x_1=1$ на адрес 1100
	0	0	0	1	0	1	0	0	0	1	1	1	Переход, если $x_2=1$ к блоку 3 ГСА по адресу 0111 (метка M2)
	0	0	1	0	1	0	0	0	1	0	0	0	Выдача $z_4=1$ (A_4)
	0	0	1	1	0	0	0	0	0	0	0	0	Возврат в 0000 безусловно к метке M0
	0	1	0	0	-	-	-	-	-	-	-	-	Не используется
	0	1	0	1	-	-	-	-	-	-	-	-	Не используется
	0	1	1	0	-	-	-	-	-	-	-	-	Не используется
M2:	0	1	1	1	1	1	0	0	0	0	0	0	Выдача $z_1=1$ (A_1)
	1	0	0	0	1	0	1	1	0	0	0	0	Выдача $z_2, z_3=1$ (A_2)

БМУ со счётчиком МК



Вот так!

