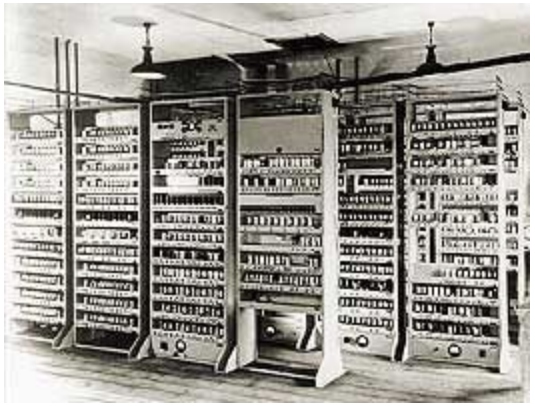
Тема 3. МИКРОПРОГРАММНЫЕ АВТОМАТЫ.

- Лекция 13.
- Синтез МПУУ на «жёсткой» логике

- 1.Автомат Уилкса. Микропрограммное устройство управления-МПУУ
- 2.ЛСА,МСА и ГСА.
- 3. Синтез МПУУ на «жёсткой» логике

1.Автомат Уилкса. Микропрограммное устройство управления.

- Дальнейшее развитие принципа «разделяй и властвуй» в дискретной технике привело к созданию микропрограммного автомата – МПА (1951 г., Кембридж, М.В. Уилкс).



Уилкс, Морис Винсент

- **Морис Винсент Уилкс (англ. *Maurice Vincent Wilkes*, 26 июня 1913 года, Дадли, Великобритания) — британский учёный в области компьютерных наук.**
- ***Профессор Уилкс более всего известен как проектировщик EDSAC - первого компьютера, допускающего внутреннее хранение программ. Построенный в 1949, EDSAC использовал память на линиях задержки. Он также известен, в соавторстве с Виллером и Гиллом как автор книги «Preparation of Programs for Electronic Digital Computers», 1951 года, в которой вводится важнейшее понятие библиотеки* программ**

Морис Уилкс

- Я точно помню тот самый момент, когда я понял, что большая часть моей жизни теперь будет состоять в поиске ошибок в моих собственных программах.

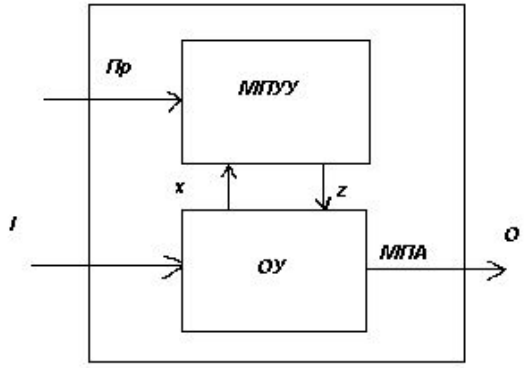
  Морис Уилкс, 1949 год

Принцип микропрограммного управления

- Пусть кто-то управляет, а кто-то – исполняет. Исполняет – операционное устройство (ОУ), где все эти АЛУ, регистры, счетчики и т.д., а управляет, реализует последовательность микрокоманд, состоящих из микроопераций, т.е. алгоритм, микропрограмму – микропрограммное устройство управления (МПУУ).

Принцип микропрограммного управления

- **МПА и МПУУ**



МПА – микропрограммный автомат

МПУУ – микропрограммное устройство управления

ОУ – операционное устройство

x – сигналы состояния ОУ- входные переменные МПУУ

z – функции управления, выходные функции МПУУ

Пр – Программа - микропрограмма

I-входная информация *O – выходная информация*

Микропрограммирование

- **Микрооперации – МО – элементарные действия обработки информации.**
- **Микрокоманды –МК – набор МО, выполняемых в одном такте.**
- **Микропрограмма – последовательность МК.**
- **Далее – Команды и Программы и т.д.**

Микропрограмма

- **Микропрограмма (англ. *firmware*, «прошивка») — программное обеспечение, встроенное («зашитое») в аппаратное устройство. Часто представляется в виде микросхем флеш-ПЗУ или в виде файлов образов микропрограммы, которые могут быть загружены в аппаратное обеспечение.**

Микропрограмма

- Программа по тактам, управляющая ресурсами вычислительного устройства (ALU, сдвигатели, мультиплексоры и др.). Обычно, в командном слове, выделяются отдельные биты для управления необходимым устройством.
- Программа конфигурирования различных ПЛИС (FPGA, CPLD, PAL и т. п.).

Программный автомат

- Потом было создано программное устройство управления, реализующее программу, потом – операционные системы и т.д. и т. п.

2. ЛСА, МСА и ГСА

- ЛСА, МСА и ГСА
- Предписание о последовательности действий алгоритма может быть представлено так называемой схемой: логической схемой алгоритма, матричной схемой алгоритма, граф-схемой алгоритма.
- Логическая схема алгоритма (ЛСА) впервые была предложена советским математиком Ляпуновым А.А. (1911-1973 гг.) в бытность его профессором кафедры математики военной артиллерийской (в те годы) академии им. Ф.Э. Дзержинского.
- ЛСА – это выражение, состоящее из символов операторов, логических условий, следующих в определенном порядке, а также нумерованных стрелок, расставленных особым образом.

ЛСА, МСА и ГСА

- **ЛСА**

$$A_0 \underset{2}{\downarrow} A_1 \bar{a} \overset{1}{\uparrow} b \overset{2}{\uparrow} A_2 \underset{3}{\downarrow} A_k \underset{1}{\downarrow} A_3 \omega \overset{3}{\uparrow}$$

МСА

- **Матричная схема алгоритма (МСА) – это квадратная матрица, элементы которой указывают условия передачи управления от i-го оператора строки к j-ому оператору столбца.**
- **Строки матрицы нумеруются от первого оператора до предпоследнего, столбцы – от второго до последнего.**

ЛСА, МСА и ГСА

- **МСА**

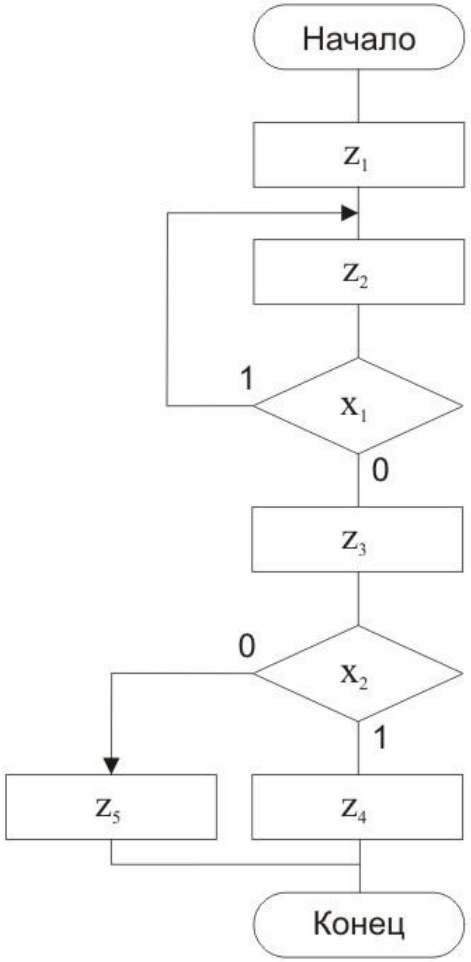
	A_1	A_2	A_3	$A_к$
A_0	1	0	0	0
A_1	$\bar{a}\bar{b}$	$\bar{a}b$	a	0
A_2	0	0	0	1
A_3	0	0	0	1

ЛСА

- **Граф-схема алгоритма (ГСА) – это ориентированный граф особого вида. Он содержит вершины четырех типов: 1) операторные, обозначаемые прямоугольниками; 2) условные, обозначаемые ромбами; 3) начальную и 4) конечную вершины, обозначаемые овалами. Вершины соединяются дугами.**

ЛСА, МСА и ГСА

- **ГСА**



Схемы алгоритмов

- **Граф-схемы алгоритмов называют просто схемами и выполняют по ГОСТ 19.701-90 «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения».**

Схемы алгоритмов

- **Начальная и конечная вершины**

- Символ отображает выход во внешнюю среду и вход из внешней среды.
- Используется для обозначения начала или окончания алгоритма.

Схемы алгоритмов

- **Линия**
- **Символ отображает поток данных или управления. Направления справа налево и снизу вверх обозначаются стрелками.**
- **Используется для соединения символов в алгоритме.**

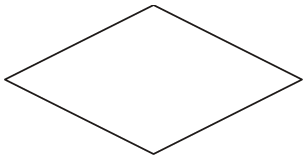
Схемы алгоритмов

- **Процесс**

- **Символ отображает функцию обработки данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации).**
- **Используется для обозначения операций присваивания.**

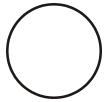
Схемы алгоритмов

- **Решение**



- **Символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа. Соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути.**
- **Используется для обозначения оператора условного перехода или оператора варианта.**

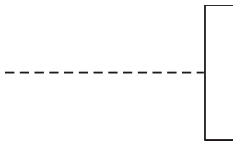
Схемы алгоритмов

- **Соединитель** 

- **Символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте. Соответствующие символы-соединители должны содержать одно и то же уникальное обозначение.**

Схемы алгоритмов

- **Комментарий**



- **Символ используется для добавления описательных комментариев или пояснительных записей с целью объяснений или примечаний. Пунктирные линии в символе комментария связаны с соответствующим символом или могут обводить группу символов. Текст комментариев или примечаний должен быть помещен около ограничивающей фигуры.**

Схемы алгоритмов

- Символы могут быть вычерчены в любой ориентации, но предпочтительной является горизонтальная ориентация.
- Внутрь символа помещают обозначения или описания операций.
- Символы могут быть отмечены идентификаторами или порядковыми номерами.
- Идентификатор представляет собой букву или букву с цифрой и должен располагаться слева над символом.
- .

Схемы алгоритмов

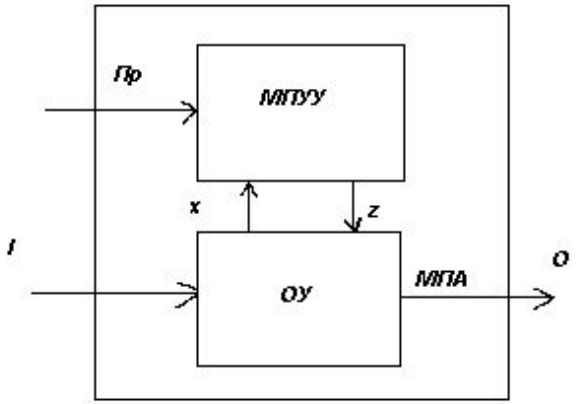
- Направления линий связи слева направо и сверху вниз считаются стандартными, и линии связи изображаются без стрелок, в противоположном случае – со стрелками.
- Линии могут соединяться одна с другой, но не могут разветвляться.

Схемы алгоритмов

- **Схема**
- **Алгоритм может быть реализован и схемой элементов (устройств), которые выполняются по ГОСТ 2.701-84 «Схемы. Виды и типы. Общие требования к выполнению».**

3.Синтез МПУУ на «жёсткой» логике

- **МПУУ**



МПА – микропрограммный автомат

МПУУ – микропрограммное устройство управления

ОУ – операционное устройство

x – сигналы состояния ОУ- входные переменные МПУУ

z – функции управления, выходные функции МПУУ

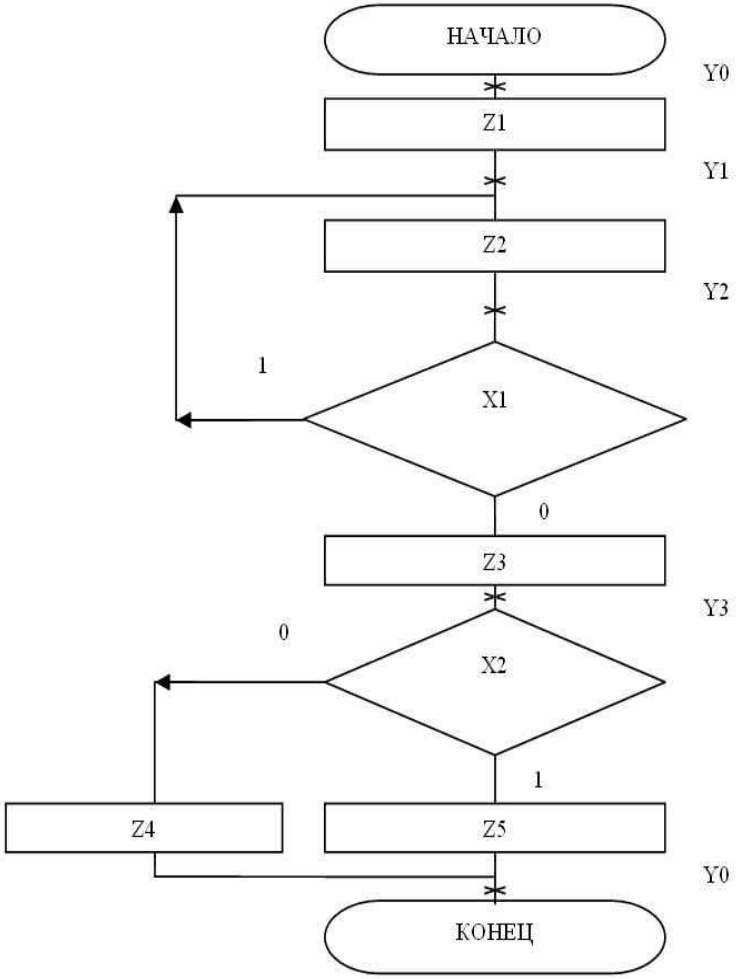
Пр – Программа - микропрограмма

I-входная информация *O – выходная информация*

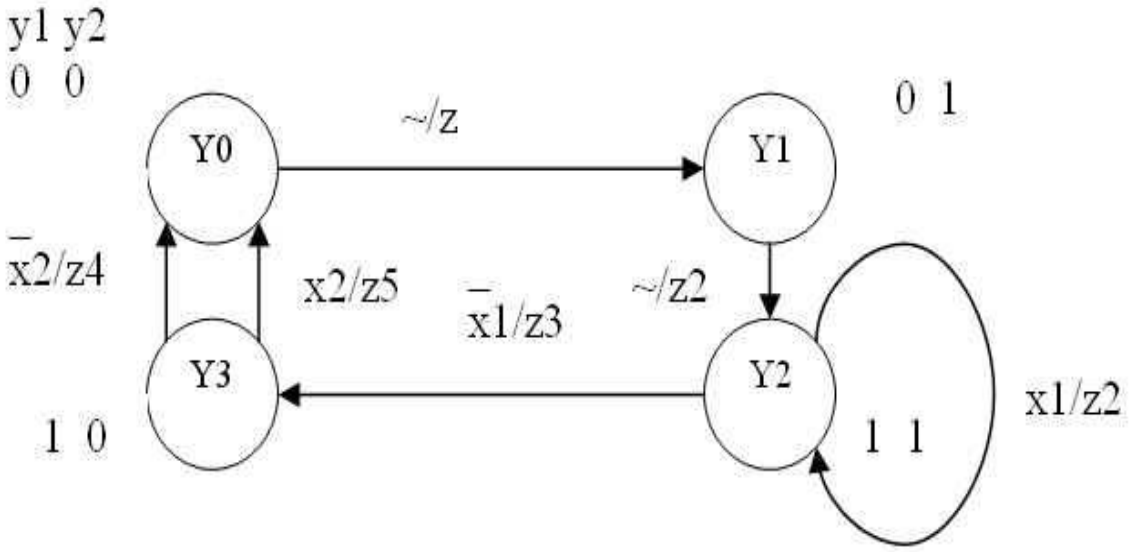
СФ

- Пусть дана следующая словесная формулировка алгоритма:
- *После начала режима подать питание, затем осуществить протяжку транспортера на 1 шаг. В случае получения сигнала «Есть продукт» – вновь осуществить протяжку на 1 шаг. Иначе – выдать сообщение «Конец продукта». Если вес продукта в норме – выдать сообщение «Работа завершена». Иначе – «Ошибка». После этого закончить работу.*
- Получим графическую схему алгоритма микропрограммного устройства управления – ГСА МПУУ

Построение ОГСА по ГСА



Построение графа управляющего автомата



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

Минимизация полученных логических функций

- **y2(t+1)=d2(t)=y1**

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

y2	y1	x2	x1
0	**1**	**~**	**~**
1	**1**	**~**	**0**
1	**1**	**~**	**1**
0	***0***	**~**	**~**
1	***0***	***0***	**~**
1	***0***	***1***	**~**

РОК

Минимизация y1

- **Импликанта (0 - - -) покрывает два рабочих набора**
- **импликанта (- 1- 1) – последний рабочий набор**

$$y1(t+1) = d1(t) = \overline{y2} \vee y1x1.$$

y2	y1	x2	x1
0	0	~	~
0	1	~	~
1	1	~	1
1	1	~	0
1	0	0	~
1	0	1	~

Минимизируем функции выходов.

- **Очевидно, что минимизация функций z1, z3, z4, z5 практически невозможна вследствие единственных их рабочих наборов. Попробуем минимизировать z2:**

$z1 = \overline{y2}\,\overline{y1};$

$z3 = y2y1\overline{x1};$

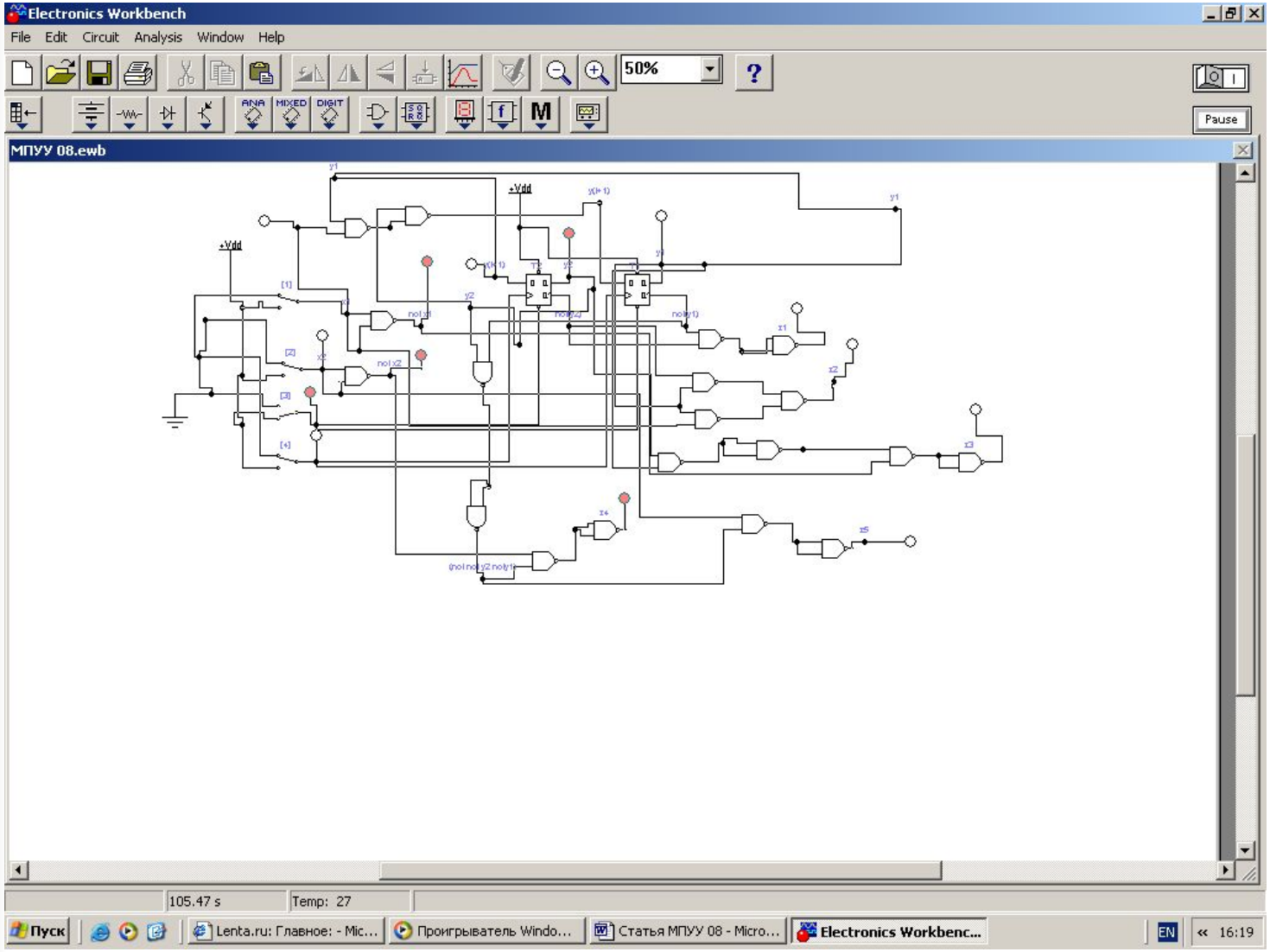
$z4 = y2\overline{y1}\,\overline{x2};$

$z5 = y2\overline{y1}x2.$

y2	y1	x2	x1
0	1	~	~
1	1	~	1
0	0	~	~
1	1	~	0
1	0	0	~
1	0	1	~

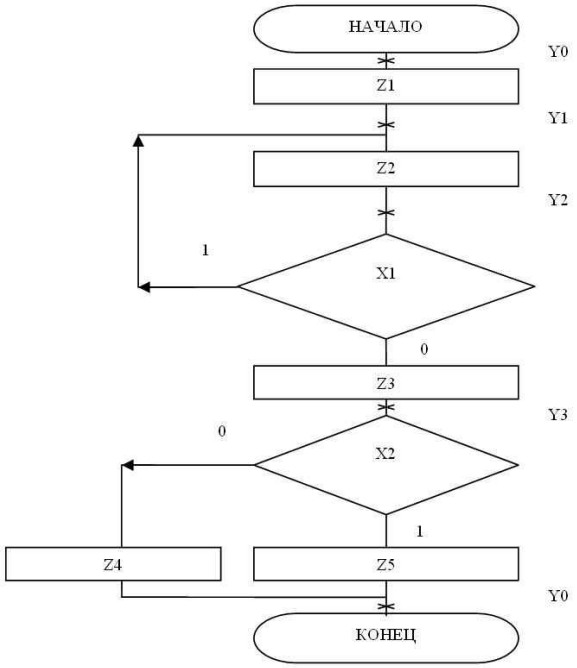
$z2 = \overline{y2}y1 \vee y1x1.$

Выполним моделирование схемы в Electronics Workbench с учётом наличия только двухвходовых элементов



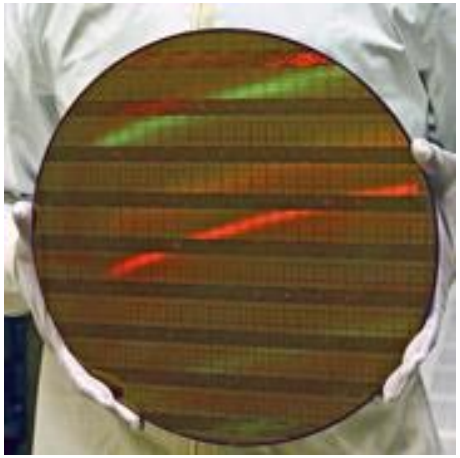
Проверка функционирования МПУУ

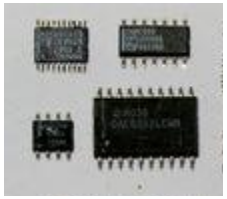
- **Итоговую проверку осуществляем по ГСА следующим образом:**
- **- устанавливаем x1=1, в этом случае последовательность выходов z1,z2,z2,...**
- **- устанавливаем x1=0, x2=1, в этом случае последовательность выходов z1,z2,z3,z5, z1,z2,z3,z5,…**
- **- устанавливаем x1=0, x2=0, в этом случае последовательность выходов z1,z2,z3,z4, z1,z2,z3,z4,…**
- **Таким образом, все три варианта выдачи последовательностей реализуются.**



МПУУ

- **Кристалл кремния**





ПЛИС


Структура ПЛИС
Конференция 2007
ПЛИС (англ. programmable logic devices – PLD)
программируемые логические интегральные схемы.
FUSION
Actel
ALTERA
XILINX
Lattice
ATMEL
Alcatel·Lucent

Процессоры и микроконтроллеры